УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Игор Гркавац

ДЕТЕРМИНАНТЕ (0,1) МАТРИЦА РЕДА ДО 8

мастер рад

Београд, 2020.

Ментор:

проф. др Миодраг Живковић, редовни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Филип Марић, ванредни професор
Универзитет у Београду, Математички факултет

др Весна Маринковић, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 28. septembar 2020.

Мојој породици

Наслов мастер рада: Детерминанте (0,1) матрица реда до 8

Резиме: Обележимо са A_n скуп свих 2^{n^2} квадратних $(0,1)$ матрица реда n. Обележимо са D_n скуп апсолутних вредности детерминанти матрица из скупа A_n. Како би се ефикасно одредио скуп D_n, полази се од подскупа скупа A_n нееквивалентних матрица таквог да је скуп апсолутних вредности детерминанти матрица из подскупа једнак D_n. За обилазак и одређивање (два различита) подскупа примењене су две методе: проширење матрица димензије $n-1$ и креирање матрица додавањем врста које испуњавају адекватне услове.

Кључне речи: матрица, детерминанта, алгоритам, група трансформација, класе еквиваленције

Предговор

Тема овог рада је ефикасно израчунавање могућих вредности детерминанти (0,1) квадратних матрица датог реда n за што веће вредности n. У енциклопедији целобројних низова [1] n-ти члан низа $A089472$ је број различитих детерминанти ових матрица за $n \leq 10$. Циљ овог рада је потврђивање вредности чланова низа за $n \leq 8$. У раду су описана два приступа проблему, њихове имплементације у језику C++, резултати извршавања, као и предлози за даљу оптимизацију.

Садржај

Глава 1

Увод

Наводе се основни појмови и тврђења о матрицама и детерминантама уопште, а затим и специјалном случају матрица над скупом $\{0, 1\}$.

1.1 Основни појмови

Матрица A величине $m \times n$ над неким скупом S представља правоугаону табелу која се састоји од m врста и n колона тако да се у пресеку i-те врсте $(1 \leq i \leq m)$ и j-те колоне $(1 \leq j \leq n)$ налази $a_{ij} \in S$. У специјалном случају када је $m = n$, A је квадратна матрица реда n.

Детерминанта квадратне матрице A реда n је скалар одређен том матрицом и записује се у следећем облику:

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

Минор M_{ij} за матрицу димензије n је детерминанта димензије $n-1$ која се добија изостављањем i-те врсте и j-те колоне. Кофактор елемента a_{ij} детерминанте $\det A$ је $A_{ij} = (-1)^{i+j} M_{ij}$.

Вредност детерминанте $\det A$ за матрицу A реда n се може израчунати применом Лапласовог развоја:

$$\det A = \sum_{i=1}^{n} a_{ij} A_{ij}$$

за произвољну колону j или

$$\det A = \sum_{j=1}^{n} a_{ij} A_{ij}$$

за произвољну врсту i. Нека се са АВД обележава апсолутна вредност детерминанте.

1.2 Особине детерминанти

Како би рачунање детерминанти било што ефикасније, потребно је избећи рачунање за оне матрице које се елементарним трансформацијама могу свести на матрице за које се већ знају вредности детерминанти. Трансформација којом се вредност детерминанте не мења је:

- Додавање једне врсте/колоне помножене неким скаларом другој врсти-/колони матрице

Трансформације којима се вредност детерминанте мења су:

- Вредност детерминанте мења знак уколико се у матрици замене места двема врстама или колонома
- Вредност детерминанте се множи скаларом s уколико се врста или колона матрице помножи скаларом s

Из претходних трансформација произилазе особине неких матрица код којих се вредност детерминанте може израчунати тривијално:

- Вредност детерминанте је 0 уколико нека врста или колона садржи само вредности 0
- Вредност детерминанте је 0 уколико су две врсте или колоне линеарно зависне

1.3 Особине $(0,1)$ матрица

$(0,1)$ матрице су матрице код којих је сваки елемент или нула или јединица. Свака $(0,1)$ матрица A реда n може се представити низом од n бројева тако да се i-та врста A_i представи бројем $\sum_{j=1}^{n} a_{ij}2^{j-1}$. Када нема опасности од забуне, за број који одговара врсти A_i може се употребити иста ознака A_i. На скупу A_n разматрају се две релације поретка:

- Прва је релација лексикографског поретка $\leq$. За две матрице A, B из скупа A_n важи да је $A \leq B$ ако важи лексикографска неједнакост истог смера за одговарајуће низове бројева. На пример:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

  јер је низ $(5,6,2)$ лексикографски мањи од низа $(6,2,3)$.

- Друга релација поретка $\preceq$ на скупу n-торки бита се може дефинисати тако што се од сваке врсте A_i формира $(n+1)$-торка A_i' уметањем на почетак броја који је једнак броју јединица у n-торки A_i. У складу са овом другом релацијом поретка важи $A_i \preceq A_j$ ако $(n+1)$-торка A_i' претходи лексикографски $(n+1)$-торци A_j'. На пример за матрицу

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

  уметањем броја јединица на почетак n-торки се добија

$$A' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

  па пошто важи $A_i' \leq A_j'$ за свако $i < j$, онда важи и $A_i \preceq A_j$ за свако $i < j$.

За две врсте матрице A кажемо да су неупоредиве ако скуп јединица једне врсте није подскуп јединица друге врсте. Упоредивост врста се може испитати преко битовског AND, где су врсте упоредиве уколико је резултат битовског AND заправо једна од две врсте. На пример, за матрицу

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

важи да су јој друга и трећа врста упоредиве, јер се битовским AND добија трећа врста.

Лема 1.3.1. *Пермутације врста и колона над матрицом А не мењају АВД те матрице.*

Доказ. Пермутација врста или колона матрице се може представити као производ операција замена места врстама или колонама те матрице. Из особине матрице да вредност детерминанте мења знак при замени места двема врстама, односно колонама, произилази да производ неког скупа трансформација замене места врста и колона, или мења знак детерминанте или вредност детерминанте остаје иста, па се самим тим АВД не мења. □

Нека Π представља групу трансформација генерисану пермутацијама врста/колона матрице. Матрице A и B су π-еквивалентне, $A \sim_\pi B$, уколико припадају истој орбити Π, односно уколико постоје пермутације врста и колона P, Q тако да важи $A = PBQ$. Тако, на пример, за матрице

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

важи $A \sim_\pi B$ јер се пермутацијом врста матрице A може добити матрица B.

π-класа еквиваленције матрице A је скуп свих матрица које су π-еквивалентне са њом. π-представник A^π матрице A је лексикографски најмања матрица из π-класе матрице A, где је π-класа матрице A орбита Π која садржи A. Тако на пример, ако је

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

онда је

$$A^{\pi} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Поред елементарних трансформација које се могу применити на матрице, а које не мењају детерминанту, над $(0, 1)$ матрицама се може дефинисати и операција ексили(XOR) X_i i-те врсте/колоне матрице A са свим осталим врстама/колонама. Уколико се ради о трансформацији врста, онда се трансформација пише са леве стране матрице, X_iA, а уколико је трансформација колона, онда се пише са десне стране, AX_i. Такође се дефинишу и идентичне трансформације $X_0A = A$ и $AX_0 = A$.

На примеру, ако је

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

онда се операцијом X_2 по врстама добија

$$X_2A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Лема 1.3.2. *Операција X_i над матрицом не мења АВД матрице.*

Доказ. Ако је A матрица реда n, онда се може дефинисати матрица $B = \Psi(A)$, која представља ± 1 матрицу реда $n + 1$, која се добија од матрице A заменом свих нула елементом -1, додавањем једне врсте са вредностима -1 изнад и додавањем једне колоне са вредностима 1 са десне стране матрице. Трансформишући матрицу A у $B = \Psi(A)$ добија се ± 1 матрица реда $n+1$. Врста i којом треба урадити трансформацију X_i матрице A је сада на $i+1$ позицији. Након замене прве и $i+1$ врсте, прва врста може да се врати на почетне вредности променом знака колона код којих је вредност елемента у првој врсти 1. Применом Ψ^{-1} на ту матрицу, се добија трансформисана матрица A, то јест A'. Дакле, трансформација X_i по врстама матрице одговара специјалној пермутацији врста у $\Psi(A)$ и множењу неких колона скаларом -1. Како те операције не мењају АВД, тако и операција X_i не мења АВД. Доказ за трансформацију X_i по колонама се изводи аналогно овом. □

Нека Φ представља групу трансформација генерисану пермутацијама врста/колона и трансформацијама X_i врста/колона. Матрице A и B су ϕ-еквивалентне, $A \sim_\phi B$, уколико припадају истој орбити Φ, односно уколико постоје пермутације врста и колона P, Q и трансформације врста и колона X_i, X_j тако да важи $A = PX_iBX_jQ$, где X_i, X_j означавају примену одговарајућих операција на врсте, односне колоне матрице. Тако, на пример, за матрице

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

важи $A \sim_\phi B$, јер се трансформацијом X_2 по колонама, а затим и пермутацијом врста, од матрице A може добити матрица B.

ϕ-представник матрице A је лексикографски најмања матрица из ϕ-класе матрице A, где је ϕ-класа матрице A орбита Φ која садржи A. Тако је на пример ϕ-представник матрице

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

матрица

$$A^{\phi} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Нека x обележава вектор колону, y вектор врсту и b скалар. Уколико матрицу A реда $n-1$ проширимо врстом (y, b) и колоном (x, b), добија се квадратна матрица реда n

$$B = bord(A, x, y, b) = \begin{bmatrix} A & x \\ y & b \end{bmatrix}$$

За проналажење свих π-представника матрице B је довољно проширити све π-представнике матрице A свим могућим комбинацијама x, y и b, а затим пронаћи π-представнике новонасталих матрица [3]. Нека је

$$bord_{\pi}(A) = \{bord_{\pi}(A, x, y, b) | x \in \{0,1\}^{n-1}, y \in \{0,1\}^{n-1}, b \in \{0,1\}\}$$

Лема 1.3.3. *Ако важи $A \sim_{\pi} A'$, онда је $bord_{\pi}(A) = bord_{\pi}(A')$.*

Доказ. Нека $B \in bord_{\pi}(A)$. Ако се пермутације врста/колона примене на првих $n-1$ врста/колона матрице B, трансформишући A у A', онда је у добијеној матрици горњи-леви минор једнак A'. Према томе, матрица која је пермутационо еквивалентна матрици B је добијена проширивањем A', што значи да је B пермутационо еквивалентна матрици из $bord(A')$, то јест $B \in bord_{\pi}(A')$. Аналогно томе, $bord_{\pi}(A') \subseteq bord_{\pi}(A)$, па је тиме $bord_{\pi}(A') = bord_{\pi}(A)$. □

На сличан начин се доказује да је за проналажење свих ϕ-представника матрице B довољно проширити све ϕ-представнике матрице A свим могућим комбинацијама x, y и b, а затим пронаћи ϕ-представнике новонасталих матрица [3]. Нека је

$$bord_{\phi}(A) = \{bord_{\phi}(A, x, y, b) | x \in \{0,1\}^{n-1}, y \in \{0,1\}^{n-1}, b \in \{0,1\}\}$$

Лема 1.3.4. *Ако важи $A \sim_{\phi} A'$, онда је $bord_{\phi}(A) = bord_{\phi}(A')$.*

Доказ. Ако су матрице A и A' ϕ-еквивалентне, онда постоји функција $g \in \Phi$ која трансформише A у A'. Нека $B \in bord_{\phi}(A)$. Тада постоји матрица $B' \in bord(A)$ таква да је $B' \sim_{\phi} B$. Применом g на горњи леви минор матрице B, добија се матрица $B'' \sim_\phi B'$, $B'' \in bord(A')$. Према томе $B \sim_\phi B''$ и $B \in bord(A')$. Пошто је матрица B ϕ-представник, добија се да је $B \in bord_\phi(A')$, што имплицира $bord_\phi(A) \subseteq bord_\phi(A')$. Аналогно томе, $bord_\phi(A') \subseteq bord_\phi(A)$, па је тиме $bord_\phi(A') = bord_\phi(A)$. $\square$

Глава 2

Алгоритми за одређивање скупа D_n матрица малог реда

Како би се смањио број израчунавања детерминанти, алгоритми који се описују у наставку се базирају на одређивању скупа представника матрица одређених класа. Да би се ово урадило што ефикасније, потребно је изабрати групу трансформација са што мањим бројем орбита и са што једноставнијом провером да ли је добијена матрица представник своје класе.

При претрази скупа свих могућих вредности детерминанте матрица реда n, може се искористити особина да вредност детерминанте мења знак при замени места двеју врста или колона, што значи да се за сваку матрицу A и израчунату вредност детерминанте d заменом двеју врста/колона матрице A зна још једна могућа вредност детерминанте и она је једнака $-d$. Према томе, рачунањем скупа D_n у који се смештају АВД, може се израчунати укупан број вредности детерминанти као $2|D_n| - 1$ (осим за специјални случај $n = 1$). До тренутка писања овога рада су познате вредности за матрице реда $n \leq 10$ [1], видети табелу 2.1.

Табела 2.1: Укупан број различитих детерминанти матрица димензије n

n	1	2	3	4	5	6	7	8	9	10
$2\|D_n\| - 1$	2	3	5	7	11	19	43	91	227	587
$\|D_n\|$	2	2	3	4	6	10	22	46	114	294

2.1 Алгоритам заснован на проширивању скупа π-представника, односно ϕ-представника

Овај алгоритам представља модификовану верзију процеса тражења детерминанти описаног у раду [3].

Из леме 1.3.3 произилази да је за проналажење свих π-представника матрице реда n довољно проширити само π-представнике матрица реда $n-1$. Након тога, потребно је израчунати апсолутне вредности детерминанте за све новодобијене матрице и сачувати у скупу D_n.
Описани алгоритам се може описати следећим псеудокодом:

Улаз : n — ред матрица за које се одређује D_n
Излаз: D_n — скуп АВД свих матрица из A_n
A_π^n — скуп свих π-представника из A_n
`// иницијализација за n=1`
$D_n \leftarrow \{0, 1\}$;
$A_\pi^1 \leftarrow \{0, 1\}$;
if $n = 1$ **then**
  return;
for $i \leftarrow 2$ **to** n **do**
  forall $A \in A_\pi^{i-1}$ **do**
    forall $x \in \{0,1\}^{i-1}$ $y \in \{0,1\}^{i-1}$ $b \in \{0,1\}$ **do**
      пронаћи π-представника $bord(A, x, y, b)$ и додати у скуп A_π^i;
forall $A \in A_\pi^n$ **do**
  израчунати АВД матрице A и додати у скуп D_n

Коректност алгоритма доказује следећа теорема:

Теорема 2.1.1. *За свако дато $n \geq 1$, алгоритам враћа све АВД и све π-представнике матрица реда n.*

Доказ. Базни случај: За $n = 1$ добијамо вредности 0 и 1, које су уједно и једине вредности за матрице реда 1.

Корак: Претпоставимо да пре уласка у петљу важи да скуп A_π^{i-1} садржи све π-представнике матрица реда $i-1$. Пролазак кроз петљу проширује све π-представнике реда $i-1$ у π-представнике реда i (видети лему 1.3.3).

Када се заврши петља по i, скуп A_π^n садржи све π-представнике матрица реда n. У наредној петљи одређује се скуп D_n. □

Да би се још више смањио број матрица које се проширују и рачунају, алгоритам се може модификовати тако да се уместо π-представника користе ϕ-представници. Модификовани алгоритам се може описати следећим псеудокодом:

Улаз : n — ред матрица за које се одређује D_n
Излаз: D_n — скуп АВД свих матрица из A_n
A_ϕ^n — скуп свих ϕ-представника из A_n

```
// иницијализација за n=1
D_n ← {0, 1};
A^1_φ ← {0, 1};
if n = 1 then
    return;
for i ← 2 to n do
    forall A ∈ A^{i-1}_φ do
        forall x ∈ {0,1}^{i-1} y ∈ {0,1}^{i-1} b ∈ {0,1} do
            пронаћи φ-представника bord(A, x, y, b) и додати у скуп A^i_φ;
forall A ∈ A^n_φ do
    израчунати АВД матрице A и додати у скуп D_n
```

Коректност алгоритма доказује следећа теорема:

Теорема 2.1.2. *За сваку дату величину n, алгоритам враћа све АВД и све ϕ-представнике матрица реда n.*

Доказ. Базни случај: За $n = 1$ добијамо вредности 0 и 1, које су уједно и једине вредности за матрице реда 1.

Корак: Претпоставимо да пре уласка у петљу важи да скуп A_ϕ^{i-1} садржи све ϕ-представнике матрица реда $i-1$. Пролазак кроз петљу проширује све ϕ-представнике реда $i-1$ у ϕ-представнике реда i (видети лему 1.3.4).

Када се заврши петља по i, скуп A_ϕ^n садржи све ϕ-представнике матрица реда n. У наредној петљи одређује се скуп D_n. □

У табели 2.2 приказано је извршавање алгоритма за претрагу π-представника за случај $n = 2$. У табели 2.3 приказано је извршавање алгоритма за претрагу ϕ-представника за случај $n = 2$. У табели 2.4 су приказани бројеви π и ϕ представника за $n \leq 6$.

2.2 Алгоритам заснован на додавању врста на правоугаоне матрице

Овај алгоритам представља модификацију алгоритма за проналажење представника латинских квадрата [4]. Алгоритам се заснива на одсецању у току претраге када се наиђе на матрицу A која се елементарним операцијама може трансформисати у матрицу A' која је већ раније обрађена, то јест $A' \leq A$. За матрицу A која се обрађује у овом алгоритму ћемо рећи да су јој врсте сортиране ако важи $A_i \leq A_j$ ако је $i < j$.

Лема 2.2.1. *Уколико је матрица A реда n π-представник, онда је њена прва врста облика $0^l 1^{n-l}$ где је $0 \leq l \leq n$.*

Доказ. Уколико се претпостави супротно, да прва врста π-представника нема облик $0^l 1^{n-l}$, тада се пермутацијом колона прва врста може свести на наведени облик, и тада је новодобијена прва врста лекскографски мања од почетне, па почетна матрица није π-представник. □

Такође, због специфичности облика прве врсте матрице, може се са l обележити позиција у првој врсти где престају вредности 0 и почињу вредности 1. Тако се број пронађених представника може смањити још више додавањем услова за врсту $k = 2$ матрице A, тако да на интервалима $[0, l-1]$ и $[l, n]$ морају бити низови нула и јединица облика [0,0,...,0,1,...,1]. Доказ се може извести аналогно као за прву врсту: пермутацијом колона би се добила врста мање лексикографске вредности, а како се све пермутације дешавају на интервалима где су елементи прве врсте међусобно једнаки, прва врста ће остати иста.

Како разматрамо алгоритме који праве списак представника за неку групу трансформација, тако се и у овом алгоритму траже представници, али се

Табела 2.2: Пример извршавања првог алгоритма за претрагу π-представника за $n = 2$

$A^1_\pi = \{ \boxed{0}, \boxed{1} \}$					
$i = 2, A = \boxed{0}$					
x	y	b	B	B^π	коментар
$[0]$	$[0]$	0	$\begin{bmatrix}0&0\\0&0\end{bmatrix}$	$\begin{bmatrix}0&0\\0&0\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
$[0]$	$[0]$	1	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
$[0]$	$[1]$	0	$\begin{bmatrix}0&0\\1&0\end{bmatrix}$	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$	без промене
$[0]$	$[1]$	1	$\begin{bmatrix}0&0\\1&1\end{bmatrix}$	$\begin{bmatrix}0&0\\1&1\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
$[1]$	$[0]$	0	$\begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$	без промене
$[1]$	$[0]$	1	$\begin{bmatrix}0&1\\0&1\end{bmatrix}$	$\begin{bmatrix}0&1\\0&1\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
$[1]$	$[1]$	0	$\begin{bmatrix}0&1\\1&0\end{bmatrix}$	$\begin{bmatrix}0&1\\1&0\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
$[1]$	$[1]$	1	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
$i = 2, A = \boxed{1}$					
x	y	b	B	B^π	коментар
$[0]$	$[0]$	0	$\begin{bmatrix}1&0\\0&0\end{bmatrix}$	$\begin{bmatrix}0&0\\0&1\end{bmatrix}$	без промене
$[0]$	$[0]$	1	$\begin{bmatrix}1&0\\0&1\end{bmatrix}$	$\begin{bmatrix}0&1\\1&0\end{bmatrix}$	без промене
$[0]$	$[1]$	0	$\begin{bmatrix}1&0\\1&0\end{bmatrix}$	$\begin{bmatrix}0&1\\0&1\end{bmatrix}$	без промене
$[0]$	$[1]$	1	$\begin{bmatrix}1&0\\1&1\end{bmatrix}$	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$	без промене
$[1]$	$[0]$	0	$\begin{bmatrix}1&1\\0&0\end{bmatrix}$	$\begin{bmatrix}0&0\\1&1\end{bmatrix}$	без промене
$[1]$	$[0]$	1	$\begin{bmatrix}1&1\\0&1\end{bmatrix}$	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$	без промене
$[1]$	$[1]$	0	$\begin{bmatrix}1&1\\1&0\end{bmatrix}$	$\begin{bmatrix}0&1\\1&1\end{bmatrix}$	без промене
$[1]$	$[1]$	1	$\begin{bmatrix}1&1\\1&1\end{bmatrix}$	$\begin{bmatrix}1&1\\1&1\end{bmatrix}$	$A^2_\pi \leftarrow A^2_\pi \cup \{B^\pi\}$
Добијени π-представници:					
$A^2_\pi = \{ \begin{bmatrix}0&0\\0&0\end{bmatrix}, \begin{bmatrix}0&0\\0&1\end{bmatrix}, \begin{bmatrix}0&0\\1&1\end{bmatrix}, \begin{bmatrix}0&1\\0&1\end{bmatrix}, \begin{bmatrix}0&1\\1&0\end{bmatrix}, \begin{bmatrix}0&1\\1&1\end{bmatrix}, \begin{bmatrix}1&1\\1&1\end{bmatrix} \},\ \lvert A^2_\pi \rvert = 7$					

Табела 2.3: Пример извршавања првог алгоритма за претрагу ϕ-представника за $n = 2$

$A^1_\phi = \{[0], [1]\}$					
$i = 2, A = [0]$					
x	y	b	B	B^ϕ	коментар
$[0]$	$[0]$	0	$\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$	$A^2_\phi \leftarrow A^2_\phi \cup \{B^\phi\}$
$[0]$	$[0]$	1	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	$A^2_\phi \leftarrow A^2_\phi \cup \{B^\phi\}$
$[0]$	$[1]$	0	$\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[0]$	$[1]$	1	$\begin{bmatrix}0 & 0\\1 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[1]$	$[0]$	0	$\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[1]$	$[0]$	1	$\begin{bmatrix}0 & 1\\0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[1]$	$[1]$	0	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	$A^2_\phi \leftarrow A^2_\phi \cup \{B^\phi\}$
$[1]$	$[1]$	1	$\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	без промене
$i = 2, A = [1]$					
x	y	b	B	B^ϕ	коментар
$[0]$	$[0]$	0	$\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[0]$	$[0]$	1	$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	без промене
$[0]$	$[1]$	0	$\begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[0]$	$[1]$	1	$\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	без промене
$[1]$	$[0]$	0	$\begin{bmatrix}1 & 1\\0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
$[1]$	$[0]$	1	$\begin{bmatrix}1 & 1\\0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	без промене
$[1]$	$[1]$	0	$\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	без промене
$[1]$	$[1]$	1	$\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	без промене
Добијени ϕ-представници:					
$A^2_\phi = \{\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\},\ \lvert A^2_\phi\rvert = 3$					

Табела 2.4: Број π и ϕ представника за $n \leq 6$

n	Број π-представника	Број ϕ-представника
1	2	2
2	7	3
3	36	12
4	317	39
5	5624	388
6	251610	8102

уместо проналажења матрице и тражења њеног представника искључују из разматања матрице које могу да се трансформишу на неку лексикографски мању од себе.

Лема 2.2.2. *Ако за матрице A и A' важи $A \sim_\pi A'$ онда су врсте у матрици A неупоредиве ако и само ако су врсте у A' неупоредиве.*

Доказ. За пермутације врста је доказ тривијалан, јер се пермутацијом врста не мења упоредивост врста. За пермутације колона, пермутовањем мењају позиције оба елемента чија се упоредивост проверава, па се упоредивост такође не мења. То значи да се π трансформацијама не мења особина неупоредивости колона па тако важи тврђење леме. □

Теорема 2.2.3. *Уместо разматрања свих π-представника, довољно је разматрати само представнике матрица којима су сваке две врсте неупоредиве и у којима прва врста има највише $\lceil n/2 \rceil$ јединица.*

Доказ. Претпоставимо да матрица A не испуњава први услов и да постоје две врсте A_i і A_j, где је једна врста подскуп друге. Одузимањем врста одређени број пута, можемо свести матрицу A на неку другу матрицу A' у којој су врсте неупоредиве. Како се сваким одузимањем смањује број јединица, процес свођења има коначан број корака. Уколико други услов није испуњен, и матрица у првој врсти има више од $\lceil n/2 \rceil$ јединица, онда се операцијом X_n по колонама матрице, добија врста чији је број јединица $\leq n/2$, чиме смо матрицу свели на неку која је већ раније обрађена. □

Алгоритам почиње додељивањем бинарне репрезентације броја првој врсти, где је вредност броја $A_1 = 2^i - 1$ $(1 \leq i \leq \lceil n/2 \rceil)$.
Затим се у петљи за $2 \leq k \leq n$ за сваку вредност броја A_{k-1} на матрицу ди-

мензија $(k-1) \times n$ додаје нова врста A_k. Свака нова врста мора да испуњава следеће услове:

1. Број јединица мора бити већи од броја јединица прве врсте, или да има исти број јединица као прва врста, а да одговарајући број буде строго већи од броја који одговара првој врсти, односно, $A_1 \preceq A_i$ где је $2 \leq i \leq n$
2. Скуп јединица у врсти мора бити неупоредив са осталим врстама
3. Проширена матрица треба да буде π-представник своје класе

Уколико су сви услови испуњени у последњој врсти матрице, матрица се додаје у скуп A_ψ. Описани алгоритам се може описати следећим псеудокодом:

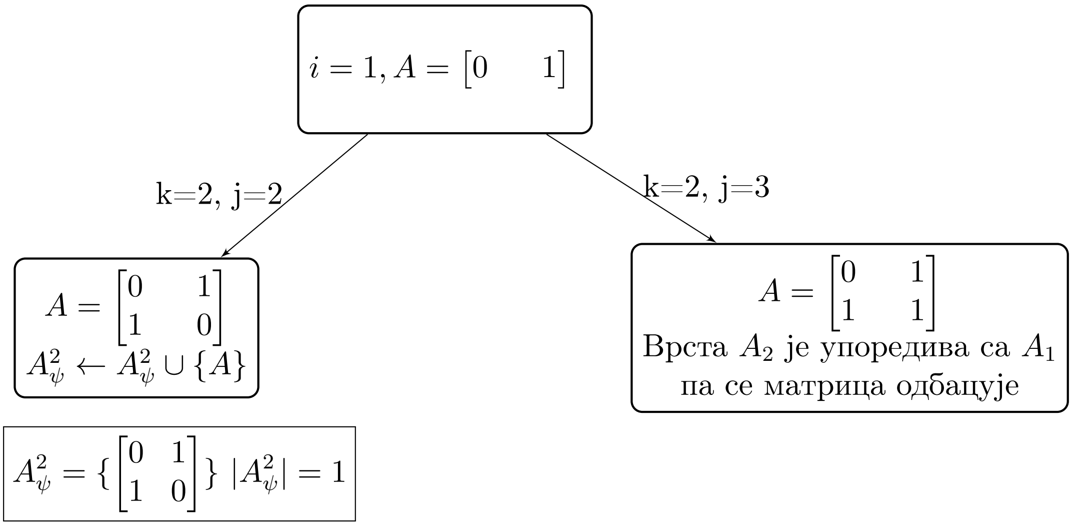
Улаз : n — ред матрица за које се одређује D_n
Излаз: D_n — скуп АВД свих матрица из A_n
A_ψ^n — скуп свих π-представника из A_n таквих да су им врсте неупоредиве
`// иницијализација скупа` π`-представника матрица реда` n
$A_\psi \leftarrow \{\}$;
$D_n \leftarrow \{\}$;
if n = *1* **then**
  $A_\psi^n \leftarrow \{\begin{bmatrix}0\end{bmatrix}, \begin{bmatrix}1\end{bmatrix}\}$;
  $D_n \leftarrow \{0, 1\}$;
  return;
for $i \leftarrow 1$ **to** $\lceil n/2 \rceil$ **do**
  $A_1 = 2^i - 1$;
  `// рекурзивни алгоритам`
  `// проширивање другом врстом`
  $matrix\_iteration(A, 2)$;
forall $A \in A_\psi^n$ **do**
  израчунати АВД матрице A и додати у скуп D_n

```
// функција matrix_iteration(A, k)
  Улаз: A — правоугаона матрица којој треба додати k-ту врсту
        k — тренутна врста коју треба додати
Излаз: A_ψ^n — скуп свих пронађених π-представника, којима су врсте
        неупоредиве, матрица A_n добијених проширивањем A
for j ← A_{k-1} + 1 to 2^n − 1 do
    A_k = j;
    // matrix_check проверава да ли су сви услови новододате
       врсте испуњени
    if matrix_check(A) = false then
        continue;
    if k = n then
        додати матрицу A у A_ψ^n;
    else
        matrix_iteration(A, k + 1);
```

У току извршавања овог алгоритма ниједан π-представник матрица код којег су сваке две врсте неупоредиве и код којег прва врста нема више од $\lceil n/2 \rceil$ јединица се не прескаче, што доказује следећа теорема.

Теорема 2.2.4. *Описани алгоритам исписује све АВД и све π-представнике правоугаоних матрица код којих су сваке две врсте неупоредиве и код којих прва врста нема више од $\lceil n/2 \rceil$ јединица.*

Доказ. Базни случај: Формира се правоугаона $1 \times n$ матрица и у њу се уписује $2^i - 1$ за $1 \leq i \leq \lceil n/2 \rceil$. За овај случај је теорема увек тачна јер је прва врста једина и испуњава све услове (видети лему 2.2.1).
Корак: Правоугаона $(k-1) \times n$ матрица A којој су сваке две врсте неупоредиве, проширује се врстом A_k тако да новонастала матрица одржава наведену особину сортираности и неупоредивости.
Уколико услов 1. није испуњен, где је број јединица мањи од прве врсте, онда се пермутацијом колона може добити врста која је лексикографски мања од прве, а затим се пермутацијом врста може добити матрица која је лексикографски мања од почетне. Уколико услов 2. није испуњен, то значи да врста може да се трансформише у врсту мање лексикографске вредности одузимањем од неке врсте с којом је та врста упоредива. Уколико услов 3. није



Слика 2.1: Дијаграм извршавања другог алгоритма за $n = 2$

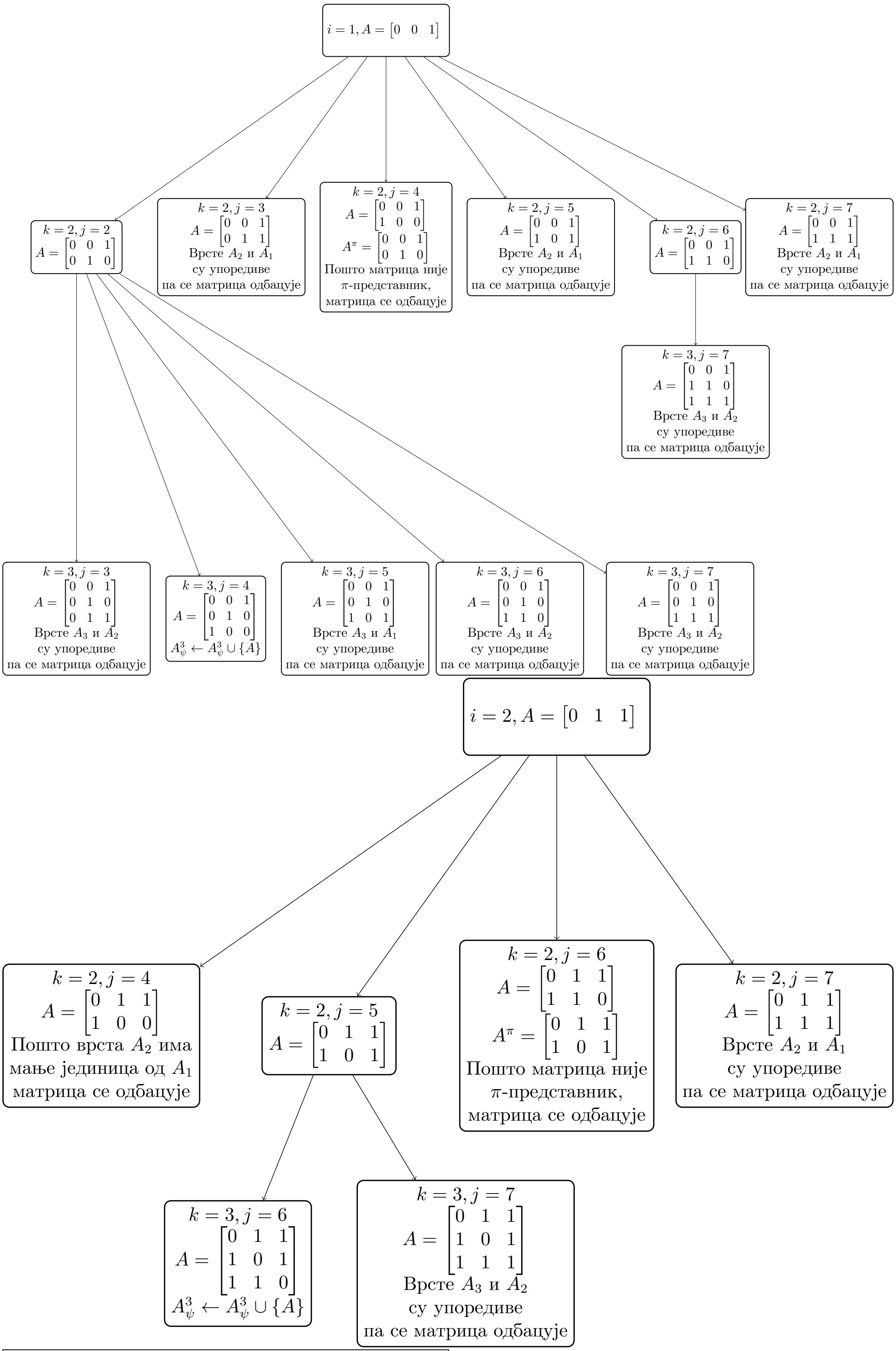
Табела 2.5: Број представника за n

n	Број π-представника са неупоредивим врстама	Број π-представника	Број ϕ-представника
1	2	2	2
2	1	7	3
3	2	36	12
4	7	317	39
5	99	5624	388
6	4593	251610	8102
7	849090	33642660	656103
8	547181566	14685630688	199727714

испуњен, онда се проналажењем π-представника матрица своди на матрицу мање лексикографске вредноси, па се та матрица такође може одбацити. $\square$

На дијаграму 2.1 је приказано извршавање алгоритма за $n = 2$. Ради бољег поређења ефикасности овог алгоритма у односу на алгоритам који се заснива на претрази π-представника и ϕ-представника, на дијаграму 2.2 је приказано извршавање алгоритма за $n = 3$.

У табели 2.5 су приказани бројеви π-представника добијени описаним алгоритмом за $n \leq 8$ и упоређени са бројевима π-представника и ϕ-представника (бројеви π-представника и ϕ-представника за $n = 7$ и $n = 8$ су преузети из [3]).



$$\boxed{A^3_\psi = \left\{ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \right\} \; |A^3_\psi| = 2}$$

Слика 2.2: Дијаграм извршавања другог алгоритма за $n = 3$

Глава 3

Програмска реализација алгоритама и резултати

Наводе се коришћене технологије и особине програмских реализација оба алгоритма, а затим и добијени резултати извршавања програма. Пореде се добијени бројеви представника и времена извршавања између имплементација оба алгоритма. У прилогу овом раду, у директоријумима *PrviAlgoritam* и *DrugiAlgoritam* су изворни кодови обе имплементације, који могу да се преведу помоћу *Cmake* алата и покрену.

3.1 Коришћене технологије

За програмску реализацију описаних алгоритама коришћене су следеће технологије:

- C++ За реализацију програма је коришћен C++ језик. Језик садржи велики број имплементираних функција и структура у стандардној библиотеци језика, при чему су неке искоришћене за трансформацију матрица у овом раду. Као стандард је изабран C++17.

- GCC Као компајлер коришћен је GCC. Главна предност за потребе овог рада јесте постављање *flag* опција за оптимизацију кода. Вредност ове опције може бити од 0 до 3, где је 0 верзија са којом је могуће користити breakpoint функције при дебаговању и где је оптимизација искључена. За потребе овог рада, изабрана је вредност 2, односно „-02”. Ова опција укључује све оптимизације „-01” које смањују величину кода и време извршавања, и такође укључује још опција за оптимизацију које повећавају перформансе кода, уз повећање и времена превођења.

- Cmake *Cmake* је алат за подешавање окружења за компајлирање програма. Помоћу њега је могуће додати листу пакета које је потребно пронаћи на рачунару како би при покретању онда *Cmake* сам пронашао потребне библиотеке. Могуће је додати и инструкције како да алат преузме одређену библиотеку са интернета и линкује је са програмом.

- Google test framework Као алат за верификацију резултата коришћен је *Google test framework*[2]. Овај алат омогућава прављење засебних тест случајева, проверавање повратних вредности функција, као и мерење времена сваког теста, без значајнијег утицаја на брзину извршавања кода.

3.2 Заједничке особине реализације оба алгоритма

Услед специфичности (0,1) матрица, резултат сваке детерминанте ће бити цео број. Самим тим, могуће је израчунати детерминанту матрице без коришћења променљивих у покретном зарезу, па је за имплементацију рачунања детерминанте искоришћен Лапласов развој (иако је сложеност већа од других начина попут *LU* декомпозиције).

3.3 Особине реализације првог алгоритма

Класа *Matrix* чува вредности матрице као низ вредности типа *bool*. У класи су као методе имплементиране претраге π, односно ϕ-представника, методе проширења матрица као и само рачунање детерминанти.

```
class Matrix{
  int size;
  std::array<bool, MAX_SIZE * MAX_SIZE> mat;
}
```

Програм прави посебну нит за сваку матрицу која се проширује и за коју се тражи π, односно ϕ-представник. Након пронађених свих представника, програм паралелно врши израчунавања АВД за сваког представника.

3.4 Особине реализације другог алгоритма

Сама матрица се чува као низ вредности типа *char*. Вектор *numbers* представља други начин чувања матрице, где је сваки елемент вектора број чија бинарна репрезентација одговара врсти матрице. Овај вектор убрзава процес испитивања упоредивости,. У скупу *solutions* се чувају вредности АВД.

```
class Matrix{
  char matrix[MAX_SIZE];
  int size;
  vector<int> numbers;
  set<int> solutions;
}
```

Након пронађених свих представника, програм покреће нити које паралелно рачунају АВД за сваког представника.

3.5 Спецификације рачунара на којем су извршена мерења

Имплементације алгоритама су покренуте на рачунару са AMD Ryzen 9 3900X процесором (12 језгара), 32GB RAM меморије. Као оперативни систем је коришћен Arch Linux (кернел верзија 5.8.8-arch1-1)

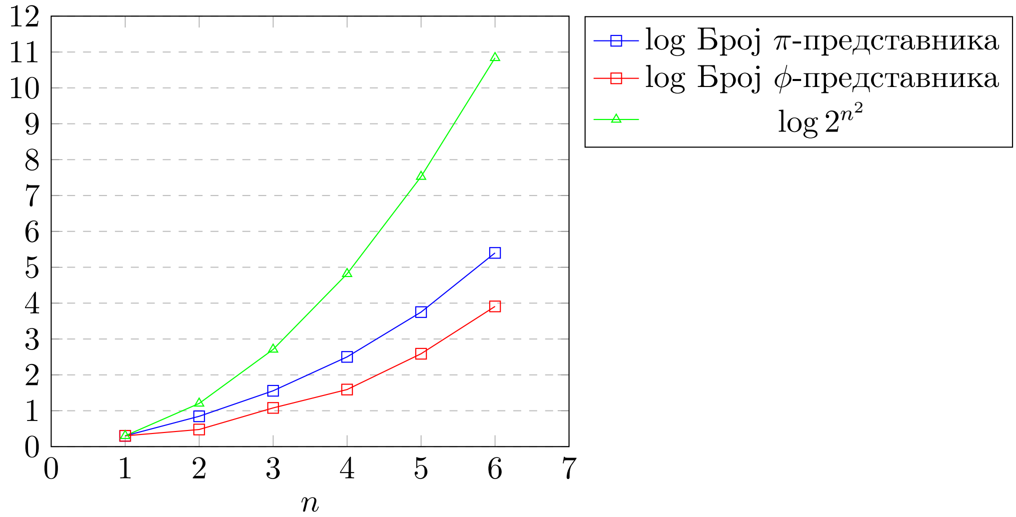
3.6 Добијени резултати

Извршавање првог алгоритма

На графику 3.1 је приказан раст пронађених π и ϕ-представника као и однос наспрам укупног броја матрица зависно од n.

У табели 3.1 су приказани бројеви пронађених π-представника, скупови D_n као и времена извршавања за одређено n.

У табели 3.2 су приказани бројеви пронађених ϕ-представника, скупови D_n као и времена извршавања за одређено n.



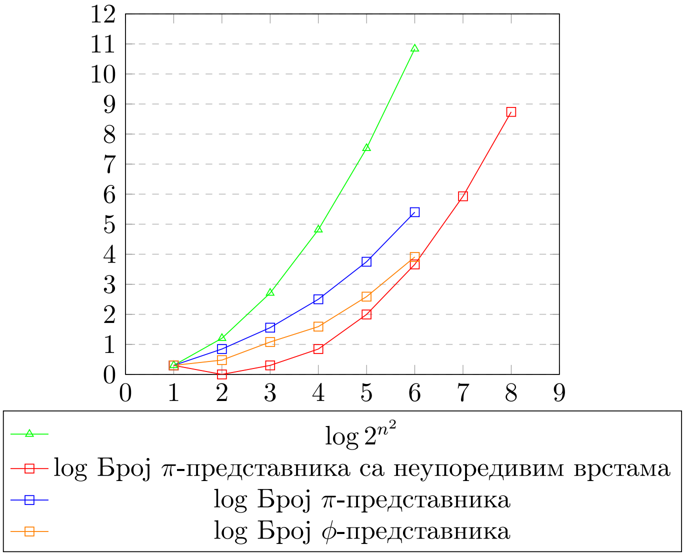
Слика 3.1: Зависност броја π, ϕ-представника и укупног броја матрица у односу на n

n	Број π-представника	$\|D_n\|$	D_n	Време извршавања (ms)
1	2	2	$\{0-1\}$	1ms
2	7	2	$\{0-1\}$	1ms
3	36	3	$\{0-2\}$	1ms
4	317	4	$\{0-3\}$	5ms
5	5624	6	$\{0-5\}$	248ms
6	251610	10	$\{0-9\}$	84880ms

Табела 3.1: Резултати извршавања првог алгоритма при претрази π-представника

n	Број ϕ-представника	$\|D_n\|$	D_n	Време извршавања (ms)
1	2	2	$\{0-1\}$	1ms
2	3	2	$\{0-1\}$	1ms
3	12	3	$\{0-2\}$	1ms
4	39	4	$\{0-3\}$	25ms
5	388	6	$\{0-5\}$	890ms
6	8102	10	$\{0-9\}$	281875ms

Табела 3.2: Резултати извршавања првог алгоритма при претрази ϕ-представника



Слика 3.2: Поређење броја π-представника са неупоредивим врстама добијених у другом алгоритму, у односу на број представника добијених првих алгоритмом и у односу на укупан број матрица у зависности од n

n	Број π-представника са неупоредивим врстама	$2\|D_n\| - 1$	$\|D_n\|$	D_n	Време извршавања (ms)
1	2	2	2	$\{0-1\}$	1ms
2	1	3	2	$\{0-1\}$	1ms
3	2	5	3	$\{0-2\}$	1ms
4	7	7	4	$\{0-3\}$	1ms
5	99	11	6	$\{0-5\}$	3ms
6	4593	19	10	$\{0-9\}$	79ms
7	849090	43	22	$\{0-18, 20, 24, 32\}$	18152ms
8	547181566	91	46	$\{0-40, 42, 44, 45, 48, 56\}$	32474381ms

Табела 3.3: Резултати извршавања другог алгоритма

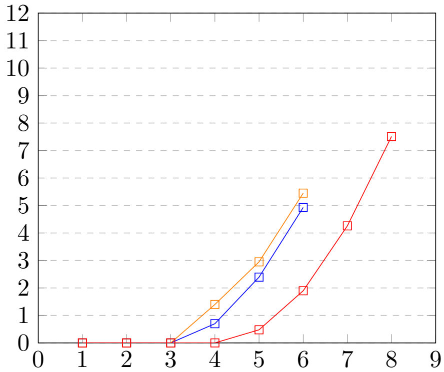
Извршавање другог алгоритма

На графику 3.2 је приказан раст пронађених π-представника са неупоредивим врстама као и однос наспрам π-представника, ϕ-представника, као и укупног броја матрица зависно од n.

У табели 3.3 су приказани бројеви пронађених π-представника са неупоредивим врстама, скупови D_n као и времена извршавања за одређено n.

На графику 3.3 је приказан однос времена извршавања оба алгоритма зависно од n.

Из табеле 3.3 се може видети да се вредности $2|D_n| - 1$ поклапају са вредностима елемената низа $A089472$ за $n \leq 8$.



Слика 3.3: Поређење времена извршавања оба алгоритма у зависности од n

Глава 4

Закључак

У раду су описана два приступа проналажењу свих детерминанти $(0,1)$ матрица реда n за $n \leq 8$. Успешно су израчунате вредности за $n \leq 8$ и измерена времена потребна за извршавање оба алгоритма. Занимљиво је да, на пример, за $n = 8$ свих 2^{64} матрица има само 46 различитих АВД.

Могуће је још више унапредити алгоритам који је заснован на додавању врста правоугаоним матрицама пооштравањем услова да ли је матрица представник додавањем услова да се добијена матрица упореди са транспонованом матрицом. Тиме се смањује број представника, јер се релација неупоредивости врста проширује и на неупоредивост колона. Смањује се и број претражених матрица за фактор два пошто су избачене све матрице код којих је транспонована матрица мања од добијене матрице.
Алгоритам за претрагу π-представника се може оптимизовати тако, да се смањи број могућих пермутација које се извршавају и проверавају.
Уколико се описани алгоритми специјализују само за проналажење представника матрица за које је потребно рачунање вредности детерминанти, без рачунања вредности детерминанте, добијени представници се могу проследити неким од програмибилних картица попут Maxeler[1] које имају брже време израчунавања над великим скупом података.

[1] http://www.maxeler.com/

Библиографија

[1] The on-line encyclopedia of integer sequences. published electronically at https://oeis.org, 2010, Sequence A089472.

[2] Google. Google test framework. on-line at: `https://github.com/google/googletest/blob/master/googletest/docs/primer.md`.

[3] Miodrag Živković. Classification of small $(0, 1)$ matrices. *Linear Algebra and its Applications 414*, pages 310–346, 2006.

[4] Владан Радивојевић. Магистарски рад, Класификација латинских квадрата, Математички факултет Београд. 2009.