Универзитет у Београду
Математички факултет

Мастер рад

Игра Ним

Аутор:

Марија Мијаиловић

Ментор:

Др Миодраг Живковић

Катедра за рачунарство и информатику

Београд, мај 2020

Садржај

1 Увод

2 Витхоф-ова игра

Витхоф-ова игра (енг. *Wythoff's game*)[3] је математичка стратешка игра за два играча. На талону су нам дате две гомиле жетона, играчи наизменично узимају жетоне са једне или обе гомиле. Приликом узимања жетона са обе гомиле, рецимо $k(>0)$ са једне и $l(>0)$ са друге број узетих жетона мора задовољити услов $|k-l|<a$, где је a било који позитиван број. Игра се завршава када број жетона на талону буде нула, а онај играч који је уклонио последњи жетон или жетоне је победник. Прослеђивање није могуће - сваки играч када је на потезу мора да уклони бар један жетон.

У класичној Витхоф игри $a=1$, што значи да ако играч узима жетоне са обе гомиле, број узетих жетона мора бити једнак.

Еквивалентни опис игре би био: Имамо једну шаховску краљицу постављену било где на табли, сваки играч може да помера краљицу произвољан број корака у правцу југа, запада, или југозапада. Победник је играч који први помери краљицу у доњи леви ћошак табле.[2] [4]

Постоје тврдње да се ова игра играла у Кини под именом "捡石子 jiǎn shízǐ"(енг. *picking stones*). [6]

Холандски математичар В. А. Витхоф (енг. *W. A. Wythoff*) је 1907. године објавио математичку анализу ове игре. [5]

3 Оптимална стратегија

Било која позиција се може представити паром бројева (a,b), где је $a \leq b$, док a и b представљају број жетона на талону или координате позиције краљице. Имамо два типа позиција око којих се врти игра, П-позиције и Н-позиције. На П-позицији, играч који је на потезу ће изгубити и са најбоље одиграним потезом, тачније претходни играч може да победи шта год одиграо противник. Док на Н-позицији, следећи играч може да победи шта год противник одиграо.

Класификација позиција на П и Н се дефинише рекурзивно на следећи начин:

1. $(0,0)$ је П-позиција јер играч који је на потезу не може да одигра ниједан валидан потез, па је његов противник победник.

2. Било која позиција са које је П-позиција достижна је Н-позиција.

3. Ако сваки потез води ка Н-позицији, онда је то П-позиција.

На пример, све позиције облика $(0,b)$ и (b,b), где је $b>0$ су Н-позиције, на основу другог правила. За $a=1$ позиција $(1,2)$ је П-позиција, зато што су са ње достижне само позиције $(0,1),(0,2),(1,0)$ и $(1,1)$, које су Н-позиције. Још неке П-позиција су $(0,0),(1,2),(3,5),(4,7),(6,10)$ и $(8,13)$.

Да би се Витхоф игра ирала на најбољи могући начин, потребно је знати две ствари:

- Препознати припроду тренутне позиције, да ли је П или Н
- Израчунати следећи потез, уколико је тренутна позиција Н

Разлог битности лежи у чињеници да уколико је тренутна позиција Н, знамо да постоји потез који нас води на П-позицију, а тај потез можемо израчунати и победити. Са друге ако је тренутна позиција П не можемо урадити ништа, само одиграти произвољан валидан потез и надати се најбољем, с обзиром на то да се у једном потезу са П-позиције стиже на Н-позицију, са које противник може да победи ако зна да израчуна П-позицију. У овом раду биће приказано како се може израчунати победничка позиција, користећи рекурзивну, алгебарску или аритметичку стратегију.

4 Рекурзивна стратегија

Рекурзивном стратегијом П-позиције добијају се рачунајући $B_n - A_n = an$. За A_n важи $A_n = mex\{A_i, B_i : i < n\}$, где mex дефинишемо као најмању вредност целог сортираног скупа, који не припада подскупу, тачније то је најмања вредност комплементарног скупа. Треба напоменути да је $mex\emptyset = 0$.

У случају да се играч помера са (A_n, B_n) позиције, и узима само жетоне са једне гомиле, тим потезом производи позицију која није облика (A_i, B_i). Уколико узима жетоне са обе гомиле такође производи потез који није облика (A_i, B_i), у супротном уколико би произведена позиција била (A_i, B_i), морало би да важи $|(B_n - B_i) - (A_n - A_i)| < a$, ако искористимо да је $B_n - A_n = an$ добијамо да треба да буде задовољено $|(n-i)a| < a$, што је тачно само ако је $i = n$, што је контрадикција.

У случају да се играч помера са позиције $(x, y), x \leq y$, позиција која није облика $(A_i, B_i), i \geq 0$. Како су A и B комплементарни скупови, може се сматрати да је $x = B_n$, или је $x = A_n$, за $n \geq 0$.

- Случај 1: $x = B_n$ онда $y = A_n$
- Случај 2: $x = A_n$, ако је $y > B_n$ онда $y = B_n$. Док у случају када је $A_n \leq y < B_n$ онда рачунамо $d = y - x, m = \lfloor \frac{d}{a} \rfloor$ и померамо се на позицију (A_m, B_m). Ово је легалан потез јер:
  1. $d = y - A_n < B_n - A_n = an$, стога $m = \lfloor \frac{d}{a} \rfloor \leq \frac{d}{a} < n$
  2. $y = A_n + d \geq A_m + am = B_m$
  3. $|(y - B_m) - (x - A_m)| = |d - am| < a$

5 Алгебарска стратегија

Алгебарском стратегијом П-позиције добијају се рачунајући $A'_n = \lfloor n\alpha \rfloor$, и $B'_n = \lfloor n\beta \rfloor$, где α и β рачунамо:

$$\alpha = \frac{2-a+\sqrt{a^2+4}}{2}\ ,\ \beta = \alpha + a$$

Где је α позитиван корен квадратне једначине $\xi^{-1} + (\xi + a)^{-1} = 1$, тако су α и β ирационални за сваки позитиван број a, и задовољавају $\alpha^{-1} + \beta^{-1} = 1$

Уочимо да је $A'_0 = 0, B'_0 = 0$ и $B'_n - A'_n = an$. Такође како су A'_n и B'_n растући низови и комплементарни, то важи још и да је $A'_n = mex\{A'_i, B'_i : i < n\}$. Што показује да је $A'_n = A_n$ и $B'_n = B_n$ за $n \geq 0$.

Тако да се надаље за игру може спроводити иста стратегија описана у 4.

6 Аритметичка стратегија

Дефинишемо p и q низове рекурзивно на следећи начин :

$$p_{-1} = 1, p_0 = {}_0, p_n = a_n p_{n-1} + p_{n-2}, (n \geq 1)$$
$$q_{-1} = 0, q_0 = 1, q_n = a_n q_{n-1} + q_{n-2}, (n \geq 1)$$

Где је $a_0, a_1, ...$ јединствени бесконачни низ природних бројева за које важи $a_0 = 1$ и $a_{1,2}, ...$, су позитивни и $a_n \neq 1$, тако да уколико је α ирационалан број можемо га представити следећим једноставним бесконачним верижним разломком:

$$\alpha = 1 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}} = [1, a_1, , a_2, , a_3, ...]$$

Из ове једнакости се може закључити да је

$$\frac{p_n}{q_n} = [1, a_1, , a_2, , a_3, ..., a_n].$$

Тачније $\frac{p_n}{q_n}$ је конвергент ирационалног броја α.

Сваки рационални број $\frac{m}{n}$се Еуклидовим алгоритмом може претворити у коначни једноставни верижни разломак.

$$m = nq + r \Rightarrow \frac{m}{n} = q + \frac{r}{n} = q + \frac{1}{\frac{n}{r}}$$

Процес се даље наставља дељењем n са r.

Потребно је увести p-систем и q-системе нумерације. У p-систему можемо записати сваки позитиван број, за који важи

$$N = \sum_{i=0}^{m} s_i p_i, 0 \leq s_i \leq a_{i+1}, s_{i+1} = a_{i+2} => s_i = 0, i \geq 0$$

Слично важи и за q-систем

$$N = \sum_{i=0}^{n} t_i q_i, 0 \leq t_0 \leq a_1, 0 \leq t_i \leq a_{i+1}, t_i = a_{i+1} => t_{i-1} = 0, i \geq 1$$

Где су p_i и q_i i-ти елементи горе дефинисаних низова p и q, приказ првих неколико бројева записаних у p и q систему, за $a_i = 2, i \geq 1$ дат је у табели 1.

Табела 1: Приказ првих неколико бројева записаних у p и q систему, за $a_i = 2, i \geq 1$

$\mathbf{q_3}$ 12	$\mathbf{q_2}$ 5	$\mathbf{q_1}$ 2	$\mathbf{q_0}$ 1		$\mathbf{p_3}$ 17	$\mathbf{p_2}$ 7	$\mathbf{p_1}$ 3	$\mathbf{p_0}$ 1	**n**
			1					1	**1**
		1	0					2	**2**
		1	1				1	0	**3**
		2	0				1	1	**4**
	1	0	0				1	2	**5**
	1	0	1				2	0	**6**
	1	1	0			1	0	0	**7**
	1	1	1			1	0	1	**8**
	1	2	0			1	0	2	**9**
	2	0	0			1	1	0	**10**
	2	0	1			1	1	1	**11**
1	0	0	0			1	1	2	**12**
1	0	0	1			1	2	0	**13**
1	0	1	0			2	0	0	**14**
1	0	1	1			2	0	1	**15**
1	0	2	0			2	0	2	**16**
1	1	0	0		1	0	0	0	**17**

Дефинисаћемо још *репрезентацију* R као $(m+1)$-торка

$$R = (d_m, d_{m-1}, ..., d_1, d_0), 0 \leq d_i \leq a_{i+1}, d_{i+1} = a_{i+2} => d_i = 0, i \geq 0.$$

Уколико у R померимо сваку цифру d_i у лево за једно место добијамо $R' = (d_m, d_{m-1}, ..., d_1, d_0, 0)$, а уколико је R репрезентација са $d_0 = 0$ онда када сваку цифру d_i померимо за једно место у десно добијамо $R'' = (d_m, d_{m-1}, ..., d_1)$

$I_p = \sum_{i=0}^{m} d_i p_i$ је *p-интерпретација* репрезентације R.

$I_q = \sum_{i=0}^{m} d_i q_i$ је *q-интерпретација* репрезентације R.

Може се приказати и веза између рецимо p-интерпретације и q-репрезентације за позитиван број k

$$I_p(R_q(k)) = I_p(d_m, d_{m-1}, ..., d_1) = n$$

На пример број $R_q(12) = 1000$, а $I_p(1000) = 17$, ово је приказано у табели 1.

У случају када смо на позицији $(x, y), 0 < x \leq y$, прво је потребно ирачунати $R_p(x)$ и проверити да ли се завршава са парним или непарним бројем нула.

Уколико се завршава са непарним бројем нула онда је $x = B_n$, тако да је победнички потез $(x, y) \to (I_p(R''_p(x)), x)$

Уколико се завршава са парним бројем нула онда је $x = A_n$, ако је $y > I_p(R'_p(x)$ победнички потез је $(x, y) \to (x, I_p(R'_p(x))$, иначе уколико је $y < I_p(R'_p(x)$ рачунамо $d = y - x, m = \lfloor \frac{d}{a} \rfloor$. Тако да уколико се сад $R_q(m)$ завршава са парним бројем нула онда је $A_m = I_p(R_q(m))$, иначе уколико се завршава непарним бројем нула $A_m = I_p(R_q(m)) - 1$. У оба случаја победнички потез је $(x, y) \to (A_m, A_m + ma)$

7 Имплементација и евалуација

За сваку стратегију извршено је мерење конструкције П табеле, резултати извршавања у милисеундама зависно од n, при фиксном $a = 2$ су приказани у табели 2. За мерење је коришћена хроно библиотека (енг. *chrono library*) [1]. Сва мерења су извршена на раучунару са следећом конфигурацијом:

CPU: Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz
RAM: Kingston 8GB 1600MHz DDR3
OS: Debian GNU/Linux 9 (stretch)
Compiler: gcc 6.3.0

У табели 3 приказане су величине парова жетона П табеле све до 10^{31}, као и одговарајуће n.

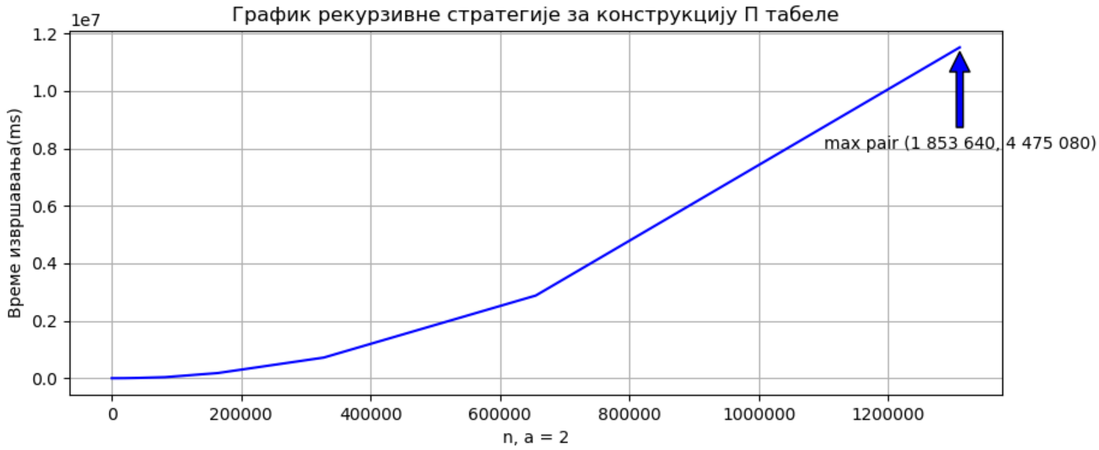
7.1 Рекурзивна стратегија

За раучунање П табеле рекурзивном стратегијом прво је потребно да израчунамо A_i, тачније потребно је наћи најмањи позитиван број који до сада није у табели - mex. За тражење је коришћен помоћни низ димензије $2 * n$, иницијализиван нулама. Тражење mex-а своди се на проналажење индекса прве нуле, с обзиром да за елементе A важи $a <= 2 * n$ сложеност у најгорем случају је $O(n)$. Чиме је укупна временска сложеност конструкције П табеле $O(n^2)$.

Listing 1: Рекурзивна стратегија рачунање П табеле

```
void Recursive::p_positions()
{
  _A.push_back(0);
  _B.push_back(0);

  vector<int>::size_type c_size = vector<int>::size_type(2*_n+1);
  _C.resize(c_size,0);

  for(int i=1;i<=_n;i++){
    int mex = get_min_positive();
    _A.push_back(mex);
    int b = _A.at(vector<int>::size_type(i))+_a*i;
    _B.push_back(b);
    _C.at(vector<int>::size_type(mex)) = mex;
    if(b <= 2*_n){
      _C.at(vector<int>::size_type(b)) = b;
    }
  }
}

int Recursive::get_min_positive()
{
  auto it = find(_C.begin()+1,_C.end(),0);
  return static_cast<int>(distance(_C.begin(), it));
}
```

Графички приказ зависности n и времена у милисекундама дат је на 7.1, за $= 2$.



7.2 Алгебарска стратегија

За рзлику од рекурзиивне стратегије која користи имплицитну рекурзију, алгебарска стратегија користи експлицитну рекурзију, рачунајући *alpha* и *beta*. Чиме је укупна временска сложеност конструкције П табеле $O(n)$.

Listing 2: Алгебарска стратегија рачунање П табеле

```
void Algebraic::p_positions()
{
  double alpha, beta;

  alpha = (2-_a+sqrt(_a*_a+4))/2;
  beta = alpha+_a;

  _A.push_back(0);
  _B.push_back(0);

  for(int i=1;i<=_n;i++){
    _A.push_back(static_cast<int>(floor(alpha*i)));
    _B.push_back(static_cast<int>(floor(beta*i)));
  }
}
```

Графички приказ зависности n и времена у милисекундама дат је на 7.2, за $= 2$.

7.3 Аритметичка стратегија

За раучунање Π табеле аритметичком стратегијом прво је потребно конструисати једноставан коначан верижни разломак, што захтева $O(n)$ времена.

Потом дефинишемо низове p и q, њихова димензија је највише $log(n)$, стога је време потребно да дефинишемо ове низове $O(log(n))$.

Преостаје још само да n бројева представимо у p и q систему, за њихово представљање у свакој итерацији имамо бинарну претрагу низова p и q којом се одређује са колико цифара треба представити број i, што је у најгорем случају једнако величини низова p и q, тачније $log(n)$. Тако да је сложеност бинарне претраге $O(log(log(n)))$. Репрезентација броја k у p или q систему се добија тако што рачунамо количник и остатак дељења броја k са одговарајућом вредности низа p или q. Уколико имамо остатак потребно је и њега представити у p или q систему, његова p или q репрезентација је позната тако да је потребно само да је прекопирамо на крај текуће p или q репрезентације броја k, не мењајући притом унапред дефинисан број цифара. Сложеност операције копирања једнака је броју елемената који се копира, што је у најгорем случају $log(k) - 1$ цифара. Како имамо n итерација укупна сложеност представљања првих n бројева у p и q систему захтева $O(n(log(log(n)) + log(n) - 1))$ времена.

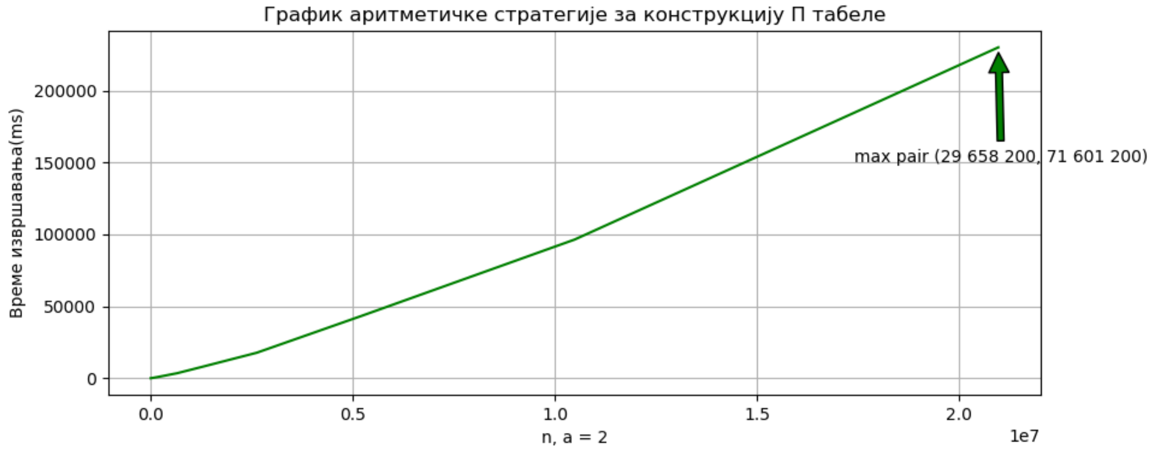
Чиме је укупна временска сложеност конструкције Π табеле $O(nlog(n))$

Listing 3: Аритметичка стратегија рачунање Π табеле

```
void Arithmetic::arithmetic_characterization_of_P_Position()
{
  alpha_continued_fractions();
  p_q_numerations();
  p_system_calculation();
  q_system_calculation();
}

void Arithmetic::alpha_continued_fractions()
{
  _alpha.push_back(1);
  fill_n(back_inserter(_alpha), _n, _a);
}

void Arithmetic::p_q_numerations()
{
  int __p = 1;
  int __q = 0;
  _p.push_back(1);
  _p.push_back(_alpha.at(1)*_p.at(0)+__p);
  _q.push_back(1);
  _q.push_back(_alpha.at(1)*_q.at(0)+__q);
  vector<int>::size_type index=2;
  int memoize = _alpha.at(2)*_p.at(1)+_p.at(0);
  while(memoize <= _n) {
    memoize = _alpha.at(index)*_p.at(index-1)+_p.at(index-2);
    _p.push_back(memoize);
    _q.push_back(_alpha.at(index)*_q.at(index-1)+_q.at(index-2));
    index++;
  }
}

void Arithmetic::p_system_calculation()
{
  vector<int>::size_type size = 0;
  int index;
  for(int i = 1; i <= _n; i++){
    int quotient = 0;
    int remainder = 0;
```

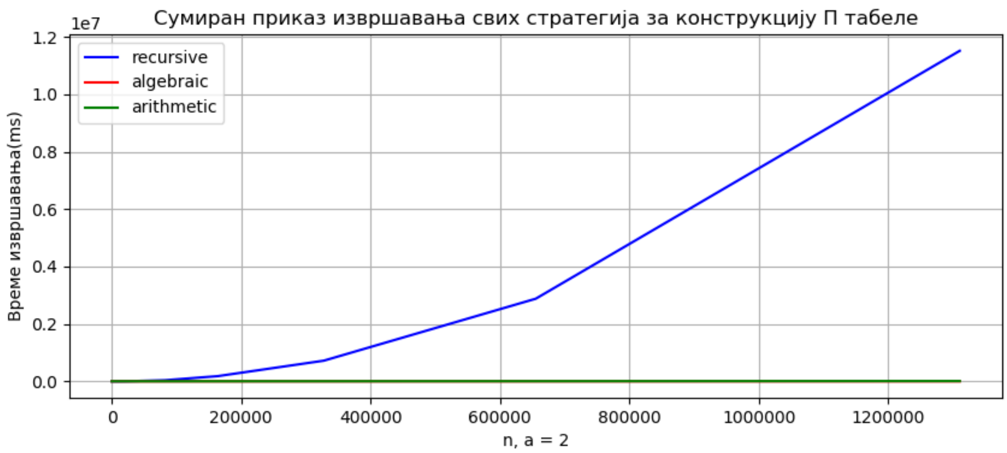
```
    //if the i is in the p, then initialize the vecor r with size zeors
    //example: i = 1, 1 is in p[0], r = {0}
    //         i = 3, 3 is in p[1], r = {0, 0}
    if(binary_search(_p.begin(), _p.end(), i)){
      size++;
      index = i;
    }
    vector<int> r(size,0);
    quotient = i/index;
    remainder = i%index;
    r.at(0) = quotient;
    if(remainder != 0){
      copy_backward(_p_system[remainder].begin(), _p_system[remainder].end(), r.end());
    }
    _p_system.insert(pair<int,vector<int>>(i, r));
  }
}

void Arithmetic::q_system_calculation()
{
  vector<int>::size_type size = 0;
  int index;
  for(int i = 1; i <= _n; i++){
    int quotient = 0;
    int remainder = 0;
    //if the i is in the q, then initialize the vecor r with size zeors
    //example: i = 1, 1 is in q[0], r = {0}
    //         i = 3, 3 is in q[1], r = {0, 0}
    if(binary_search(_q.begin(), _q.end(), i)){
      size++;
      index = i;
    }
    vector<int> r(size,0);
    quotient = i/index;
    remainder = i%index;
    r.at(0) = quotient;
    if(remainder != 0){
      copy_backward(_q_system[remainder].begin(), _q_system[remainder].end(), r.end());
    }
    _q_system.insert(pair<int,vector<int>>(i, r));
  }
}
```

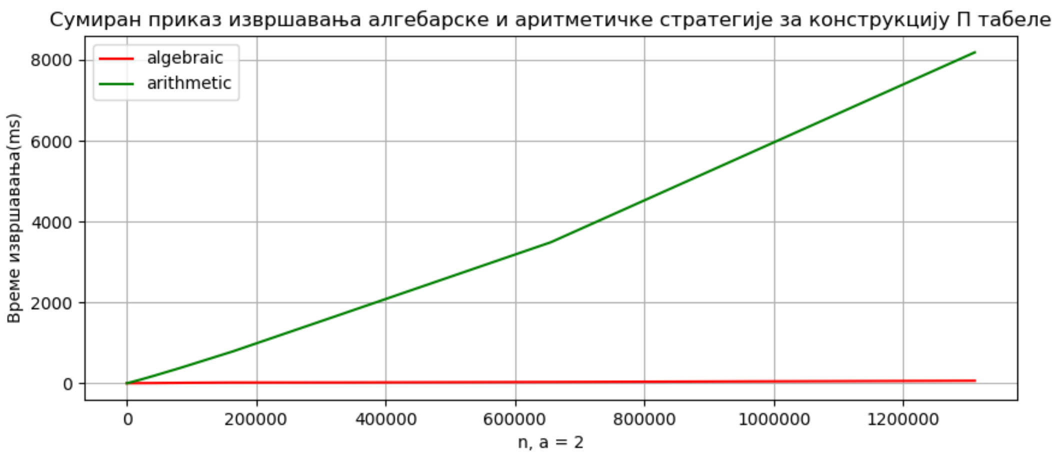
Графички приказ зависности n и времена у милисекундама дат је на 7.3, за $= 2$.



7.4 Сумиран приказ времена извршаваљњ свих стратегија

Из претходне анализе се може закључити да је алгебарска стратегија најефикаснија, што се може видети и на обједињеним графицима 7.4 7.4.





A Додатак резултатима

Табела 2: Времена извршавања конструкције П табеле

n	recursive	algebraic	arithmetic
10	0.009777	0.0063	0.036332
20	0.012459	0.005022	0.066619
40	0.024701	0.006737	0.136543
80	0.068748	0.009223	0.23613
160	0.181111	0.01631	0.471463
320	0.52075	0.025514	1.00452
640	1.88372	0.042335	2.06148
1280	7.33737	0.112218	4.33702
2560	29.0674	0.211089	9.17149
5120	116.413	0.241351	19.6074
10240	455.484	0.699545	41.4629
20480	2295.57	1.98725	84.9565
40960	10376.3	3.65746	179.124
81920	35663.2	8.41751	374.755
163840	179503	16.7582	786.699
327680	718040	17.85	1684.46
655360	2.87691e+06	30.3433	3487.31
1310720	1.1513e+07	60.9543	8186.19
2621440		133.61	17698.4
5242880		390.024	43597.4
10485760		1104.32	96394.5
20971520		2335.12	230089
41943040		3898.76	
83886080		5283.57	
167772160		12495.3	
335544320		24782.3	
671088640		42270.9	

Табела 3: Парови жетона П табеле

n	A	B
10	14	34
20	28	68
40	56	136
80	113	273
160	226	546
320	452	1092
640	905	2185
1280	1810	4370
2560	3620	8740
5120	7240	17480
10240	14481	34961
20480	28963	69923
40960	57926	139846
81920	115852	279692
163840	231704	559384
327680	463409	1.11877e+06
655360	926819	2.23754e+06
1.31072e+06	1.85364e+06	4.47508e+06
2.62144e+06	3.70728e+06	8.95016e+06
5.24288e+06	7.41455e+06	1.79003e+07
1.04858e+07	1.48291e+07	3.58006e+07
2.09715e+07	2.96582e+07	7.16012e+07
4.1943e+07	5.93164e+07	1.43202e+08

n	A	B
8.38861e+07	1.18633e+08	2.86405e+08
1.67772e+08	2.37266e+08	5.7281e+08
3.35544e+08	4.74531e+08	1.14562e+09
6.71089e+08	9.49063e+08	2.29124e+09
1.34218e+09	1.89813e+09	4.58248e+09
2.68435e+09	3.79625e+09	9.16496e+09
5.36871e+09	7.5925e+09	1.83299e+10
1.07374e+10	1.5185e+10	3.66598e+10
2.14748e+10	3.037e+10	7.33197e+10
4.29497e+10	6.074e+10	1.46639e+11
8.58993e+10	1.2148e+11	2.93279e+11
1.71799e+11	2.4296e+11	5.86557e+11
3.43597e+11	4.8592e+11	1.17311e+12
6.87195e+11	9.7184e+11	2.34623e+12
1.37439e+12	1.94368e+12	4.69246e+12
2.74878e+12	3.88736e+12	9.38492e+12
5.49756e+12	7.77472e+12	1.87698e+13
1.09951e+13	1.55494e+13	3.75397e+13
2.19902e+13	3.10989e+13	7.50794e+13
4.39805e+13	6.21978e+13	1.50159e+14
8.79609e+13	1.24396e+14	3.00317e+14
1.75922e+14	2.48791e+14	6.00635e+14
3.51844e+14	4.97582e+14	1.20127e+15
7.03687e+14	9.95164e+14	2.40254e+15
1.40737e+15	1.99033e+15	4.80508e+15
2.81475e+15	3.98066e+15	9.61016e+15
5.6295e+15	7.96131e+15	1.92203e+16
1.1259e+16	1.59226e+16	3.84406e+16
2.2518e+16	3.18453e+16	7.68813e+16
4.5036e+16	6.36905e+16	1.53763e+17
9.0072e+16	1.27381e+17	3.07525e+17
1.80144e+17	2.54762e+17	6.1505e+17
3.60288e+17	5.09524e+17	1.2301e+18
7.20576e+17	1.01905e+18	2.4602e+18
1.44115e+18	2.0381e+18	4.9204e+18
2.8823e+18	4.07619e+18	9.8408e+18
5.76461e+18	8.15239e+18	1.96816e+19
1.15292e+19	1.63048e+19	3.93632e+19
2.30584e+19	3.26095e+19	7.87264e+19
4.61169e+19	6.52191e+19	1.57453e+20
9.22337e+19	1.30438e+20	3.14906e+20
1.84467e+20	2.60876e+20	6.29811e+20
3.68935e+20	5.21753e+20	1.25962e+21
7.3787e+20	1.04351e+21	2.51924e+21
1.47574e+21	2.08701e+21	5.03849e+21
2.95148e+21	4.17402e+21	1.0077e+22
5.90296e+21	8.34804e+21	2.0154e+22
1.18059e+22	1.66961e+22	4.03079e+22
2.36118e+22	3.33922e+22	8.06158e+22
4.72237e+22	6.67843e+22	1.61232e+23
9.44473e+22	1.33569e+23	3.22463e+23
1.88895e+23	2.67137e+23	6.44927e+23
3.77789e+23	5.34275e+23	1.28985e+24
7.55579e+23	1.06855e+24	2.57971e+24
1.51116e+24	2.1371e+24	5.15941e+24
3.02231e+24	4.2742e+24	1.03188e+25
6.04463e+24	8.5484e+24	2.06377e+25

n	A	B
1.20893e+25	1.70968e+25	4.12753e+25
2.41785e+25	3.41936e+25	8.25506e+25
4.8357e+25	6.83872e+25	1.65101e+26
9.67141e+25	1.36774e+26	3.30202e+26
1.93428e+26	2.73549e+26	6.60405e+26
3.86856e+26	5.47097e+26	1.32081e+27
7.73713e+26	1.09419e+27	2.64162e+27
1.54743e+27	2.18839e+27	5.28324e+27
3.09485e+27	4.37678e+27	1.05665e+28
6.1897e+27	8.75356e+27	2.1133e+28
1.23794e+28	1.75071e+28	4.22659e+28
2.47588e+28	3.50142e+28	8.45318e+28
4.95176e+28	7.00285e+28	1.69064e+29
9.90352e+28	1.40057e+29	3.38127e+29
1.9807e+29	2.80114e+29	6.76255e+29
3.96141e+29	5.60228e+29	1.35251e+30
7.92282e+29	1.12046e+30	2.70502e+30
1.58456e+30	2.24091e+30	5.41004e+30
3.16913e+30	4.48182e+30	1.08201e+31
6.33825e+30	8.96364e+30	2.16401e+31

Литература

[1] std::chrono library. https://en.cppreference.com/w/cpp/chrono.

[2] Alexander Bogomolny. Wythoff's nim. https://www.cut-the-knot.org/pythagoras/withoff.shtml.

[3] Aviezri S. Fraenkel. How to beat your wythoff games' opponent on three fronts. *The American Mathematical Monthly*, 89(6):353–361, 1982.

[4] James Grime. Wythoff's game (get home). https://www.youtube.com/watch?v=AYOB-6wyK_I.

[5] Willem A Wythoff. A modification of the game of nim. *Nieuw Arch. Wisk*, 7(2):199–202, 1907.

[6] A. M. Yaglom and I. M. Yaglom. *Challenging Mathematical Problems with Elementary Solutions.* Holden-Day, USA, 1967.