Úvod

kuko

20.2.2018

Vybrané partie z dátových štruktúr

Administratíva

- kuko@ksp.sk
- https://people.ksp.sk/∼kuko/ds

Body

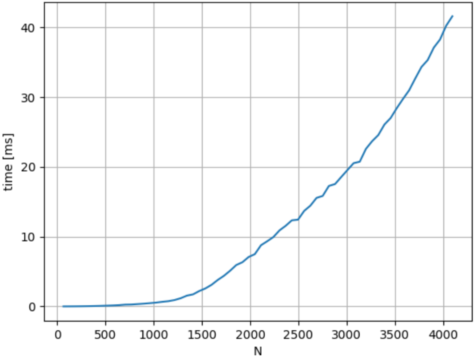
Hodnotenie: 20 bodov za D.Ú. + 20 bodov zo skúšky.

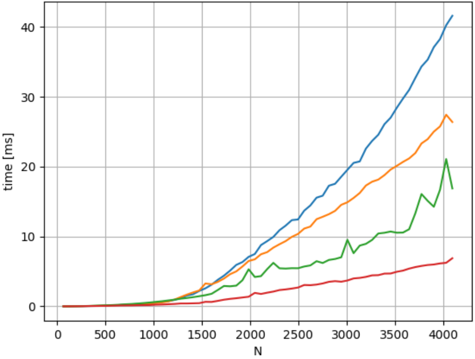
Počet bodov	Hodnotenie
40 — 35 bodov	A
34 — 29 bodov	B
28 — 23 bodov	C
22 — 18 bodov	D
17 — 12 bodov	E
< 12 bodov	Fx

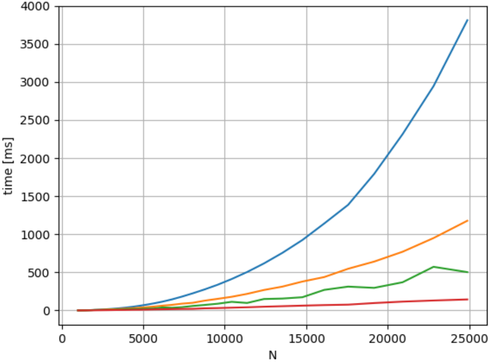
Najlacnejšia kostra a Union Find

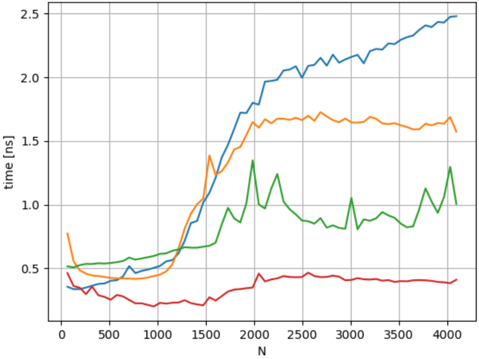
Transponovanie matice

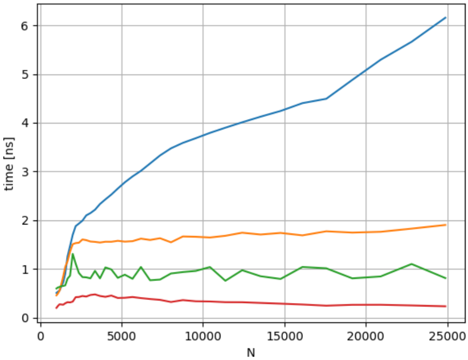
$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}^{\mathrm{T}}$$

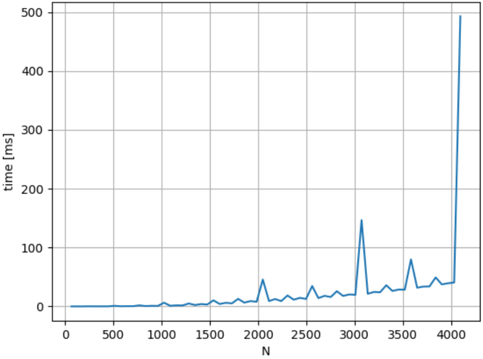

40
30
20
10
0
time [ms]
0
500
1000
1500
2000
2500
3000
3500
4000
N

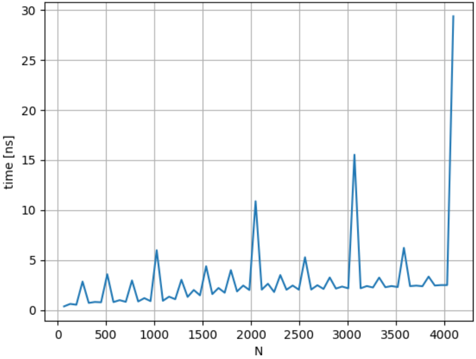

40
30
20
10
0
time [ms]
0
500
1000
1500
2000
2500
3000
3500
4000
N


4000
3500
3000
2500
2000
1500
1000
500
0
time [ms]
0
5000
10000
15000
20000
25000
N

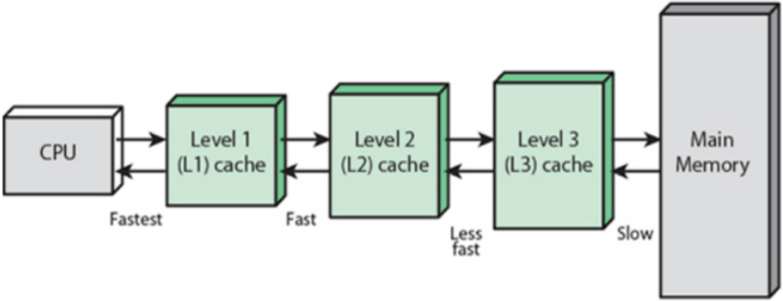

2.5
2.0
1.5
1.0
0.5
time [ns]
0
500
1000
1500
2000
2500
3000
3500
4000
N

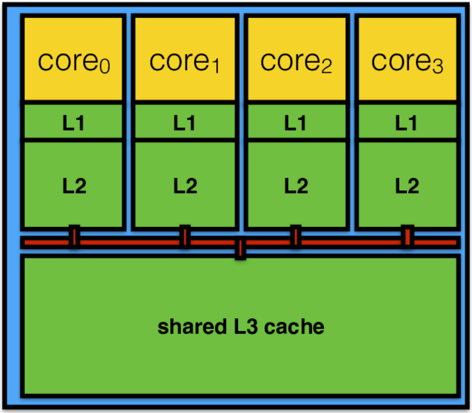

time [ns]
0
1
2
3
4
5
6
0
5000
10000
15000
20000
25000
N

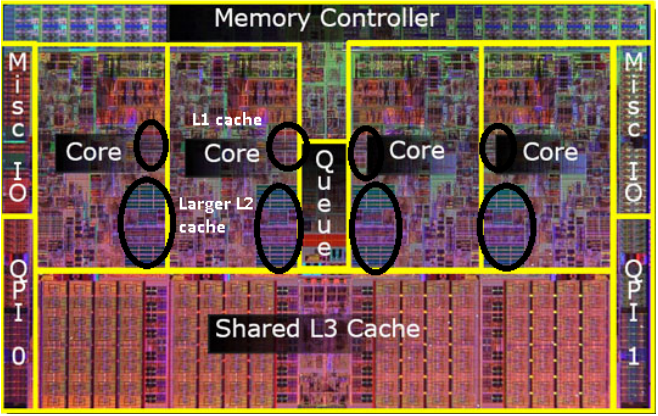

time [ms]
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500
4000
N


30
25
20
15
10
5
0
time [ns]
0
500
1000
1500
2000
2500
3000
3500
4000
N

```
void transpose_row(vector<vector<int>> &m) {
  int N = m.size();
  for (int i=0; i<N; ++i)
    for (int j=i+1; j<N; ++j)
      swap(m[i][j], m[j][i]);
}
```


CPU
Level 1 (L1) cache
Level 2 (L2) cache
Level 3 (L3) cache
Main Memory
Fastest
Fast
Less fast
Slow

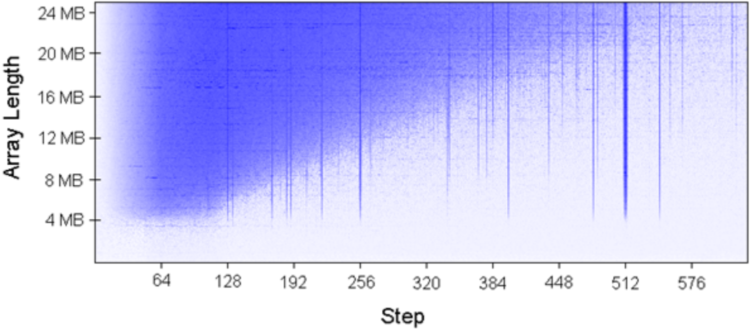

core$_0$
core$_1$
core$_2$
core$_3$
L1
L1
L1
L1
L2
L2
L2
L2
shared L3 cache

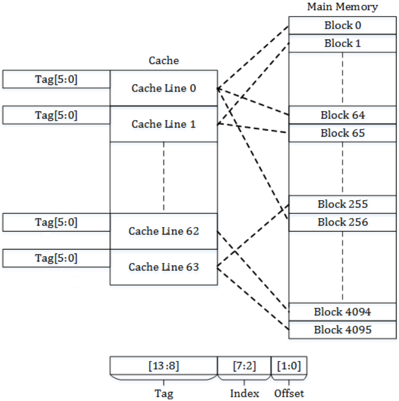

Memory Controller
Misc IO
Misc IO
Core
Core
Core
Core
L1 cache
Larger L2 cache
Queue
QPI 0
QPI 1
Shared L3 Cache

```
void transpose_block(vector<vector<int>> &m) {
  int N = m.size();
  int B = 64;
  for (int k=0; k<N; k+=B) {
    for (int i=k; i<k+B && i<N; ++i)
      for (int j=k+1; j<k+B && j<N; ++j)
        swap(m[i][j], m[j][i]);
    for (int l=k+B; l<N; l+=B)
      for (int i=k; i<k+B && i<N; ++i)
        for (int j=l; j<l+B && j<N; ++j)
          swap(m[i][j], m[j][i]);
  }
}
```

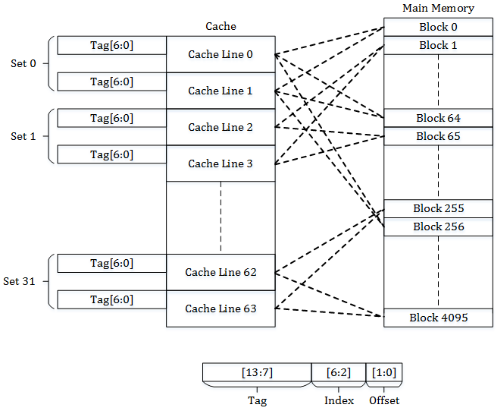
```
void transpose_block2(vector<vector<int>> &m) {
  int N = m.size();
  int B = 4;
  int B2 = 1040;
  for (int x=0; x<N; x+=B2) {
    for (int k=x; k<x+B2 && k<N; k+=B) {
      for (int i=k; i<k+B && i<N; ++i)
        for (int j=k+1; j<k+B && j<N; ++j)
          swap(m[i][j], m[j][i]);
      for (int l=k+B; l<x+B2 && l<N; l+=B)
        for (int i=k; i<k+B && i<N; ++i)
          for (int j=l; j<l+B && j<N; ++j)
            swap(m[i][j], m[j][i]);
    }
    for (int y=x+B2; y<N; y+=B2)
      for (int k=x; k<x+B2 && k<N; k+=B)
        for (int l=y; l<y+B2 && l<N; l+=B)
          for (int i=k; i<k+B && i<N; ++i)
            for (int j=l; j<l+B && j<N; ++j)
              swap(m[i][j], m[j][i]);
  }
}
```

```
void transpose_rec(const int N, vector<vector<int>> &m,
                   int n, int i0, int j0, int i1, int j1) {
  if (n <= 4) {
    if (i0==j0) {
      for (int i=i0; i<i0+n && i<N; ++i)
        for (int j=i+1; j<i0+n && j<N; ++j)
          swap(m[i][j], m[j][i]);
    } else {
      for (int i=i0, jj=j1; i<i0+n && i<N && jj<N; ++i, ++jj)
        for (int j=j0, ii=i1; j<j0+n && j<N && ii<N; ++j, ++ii)
          swap(m[i][j], m[ii][jj]);
    }
  } else {
    int h = n/2;
    transpose_rec(N, m, h, i0, j0, i1, j1);
    transpose_rec(N, m, n-h, i0+h, j0+h, i1+h, j1+h);
    transpose_rec(N, m, n-h, i0+h, j0, i1, j1+h);
  }
}
transpose_rec(N, m, N, 0, 0, 0, 0);
```


Array Length
24 MB
20 MB
16 MB
12 MB
8 MB
4 MB
64
128
192
256
320
384
448
512
576
Step



Memory Size = 16Kbytes
Memory Block Size = 4 bytes
Cache Size = 256 bytes
Block Size = 4 bytes
Associativity = 1
Number of Sets = 64



Cache

Main Memory

Set 0

Set 1

Set 31

Tag[6:0]

Cache Line 0

Cache Line 1

Cache Line 2

Cache Line 3

Cache Line 62

Cache Line 63

Block 0

Block 1

Block 64

Block 65

Block 255

Block 256

Block 4095

[13:7] [6:2] [1:0]

Tag Index Offset

Memory Size = 16Kbytes
Memory Block Size = 4 bytes
Cache Size = 256 bytes
Block Size = 4 bytes
Associativity = 2
Number of Sets = 32