

# Sample Solutions

**CTU Open Contest 2010**

# Arbitrage



# Arbitrage

- **Grafový problém**
- **Nejkratší cesta mezi libovolnými měnami**
  - Místo sčítání kombinuju násobením
- **Algoritmus Floyd-Warshall**

# Arbitrage – řešení

```
for (int k = 0 ; k < cur; ++k)
  for (int i = 0; i < cur; ++i)
    for (int j = 0; j < cur; ++j)
      D[i][j] = max(D[i][j], D[i][k]*D[k][j]);
```

- **Nepřesnosti typu double**
  - Ideálně reprezentace čísel rozkladem
  - ... nebo počítám s logaritmy
  - ... atd.

# Bus Schedule

**LIGNE 273 AGDE - LE CAP D'AGDE - AGDE - CIRCULE DU LUNDI AU DIM**

LIGNE 275

N° de Ligne	273	275	273	275	273	275	273	275
GARE SNCF	6:55	7:05	7:50	8:05	8:31	8:51	9:16	10:00
AGDE - PROMENADE - COMMISSARIAT	6:57	7:07	7:52	8:07	8:33	8:53	9:18	10:02
LA TREILLE - BALDY	6:40	-	7:55	-	8:40	-	9:40	-
CASSIN - COLLEGE - LES CAYRETS	6:43	-	7:58	-	8:43	-	9:43	-
ESPACE GRAND CAP	6:45	7:13	8:00	8:13	8:45	9:13	9:45	10:13
PUIT DE S0 - SAINT MARTIN	6:48	-	8:03	-	8:48	-	9:48	-
LES CAMPINGS DE ROCHELONGUE *	-	7:17	-	8:17	-	9:17	-	10:17
MAIL DE ROCHELONGUE - LES ONDINES	-	7:19	-	8:19	-	9:19	-	10:19
SOLEIL VILLAGE - LES TENILLES	-	7:22	-	8:22	-	9:22	-	10:22
LES NAUTILES - MONT ST MARTIN - CENTRE INFO	-	7:24	-	8:24	-	9:24	-	10:24
MUSEE DE LA CLAPE	6:52	7:26	8:07	8:26	8:52	9:26	9:52	10:26
PALAIS DES CONGRES - ZONE TECHNIQUE	6:54	-	8:09	-	8:54	-	9:54	-
ILES DES PECHEURS - LA CROISSETTE	6:56	-	8:11	-	8:56	-	9:56	-
LA GRANDE CONQUE - LE MOLE	6:58	-	8:13	-	8:58	-	9:58	-
LA CLAPE - LES AGATHES - LA ROQUILLE - CCAS - ST LOUP	7:03	-	8:18	-	9:03	-	10:03	-
CAP DE VILLAGE NATURISTE	7:08	7:31	8:24	8:31	9:08	9:31	10:08	10:31
LES ARENES	7:16	7:38	8:32	8:38	9:16	9:38	10:16	10:38
MUSEE DE LA CLAPE	7:18	7:39	8:34	8:39	9:18	9:39	10:18	10:39
CENTRE INFO - MONT ST MARTIN - LES NAUTILES	-	7:41	-	8:41	-	9:41	-	10:41
LES TENILLES - SOLEIL VILLAGE	-	7:43	-	8:43	-	9:43	-	10:43
LES ONDINES - LE MAIL DE ROCHELONGUE	-	7:46	-	8:46	-	9:46	-	10:46
LES CAMPINGS DE ROCHELONGUE *	-	7:49	-	8:49	-	9:49	-	10:49
ST MARTIN - PUIT DE S0	7:21	-	8:37	-	9:21	-	10:21	-
COLLEGE - LES CAYRETS	7:23	-	8:39	-	9:23	-	10:23	-
ESPACE GRAND CAP	7:25	7:52	8:41	8:52	9:25	9:52	10:25	10:52
CASSIN - BALDY - LA TREILLE	7:30	-	8:46	-	9:30	-	10:30	-
COMMISSARIAT - PROMENADE	7:33	-	8:49	-	9:33	-	10:33	-
GARE SNCF	7:55	8:00	8:50	9:00	9:35	10:00	10:40	11:00

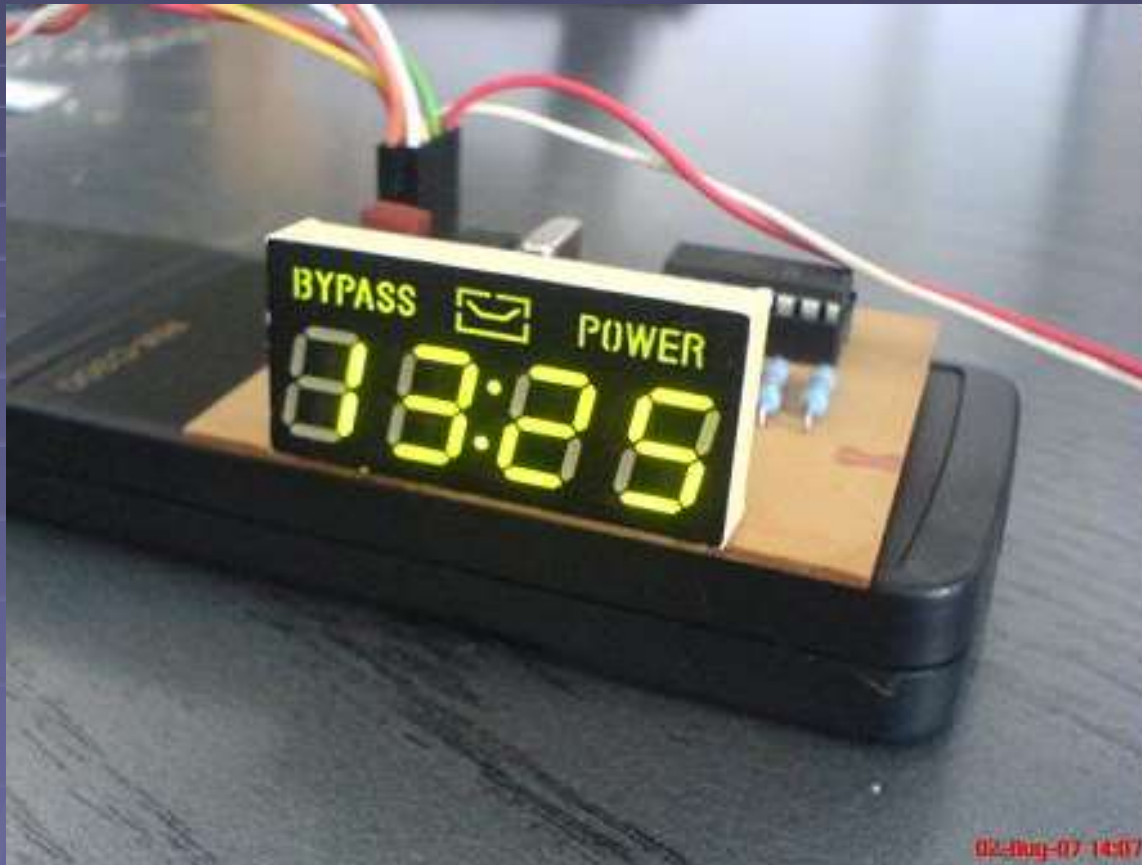
- Arrêt non desservi \*arrêts desservis

AGDE - LE GRAU D'AGDE - AGDE

# Bus – řešení

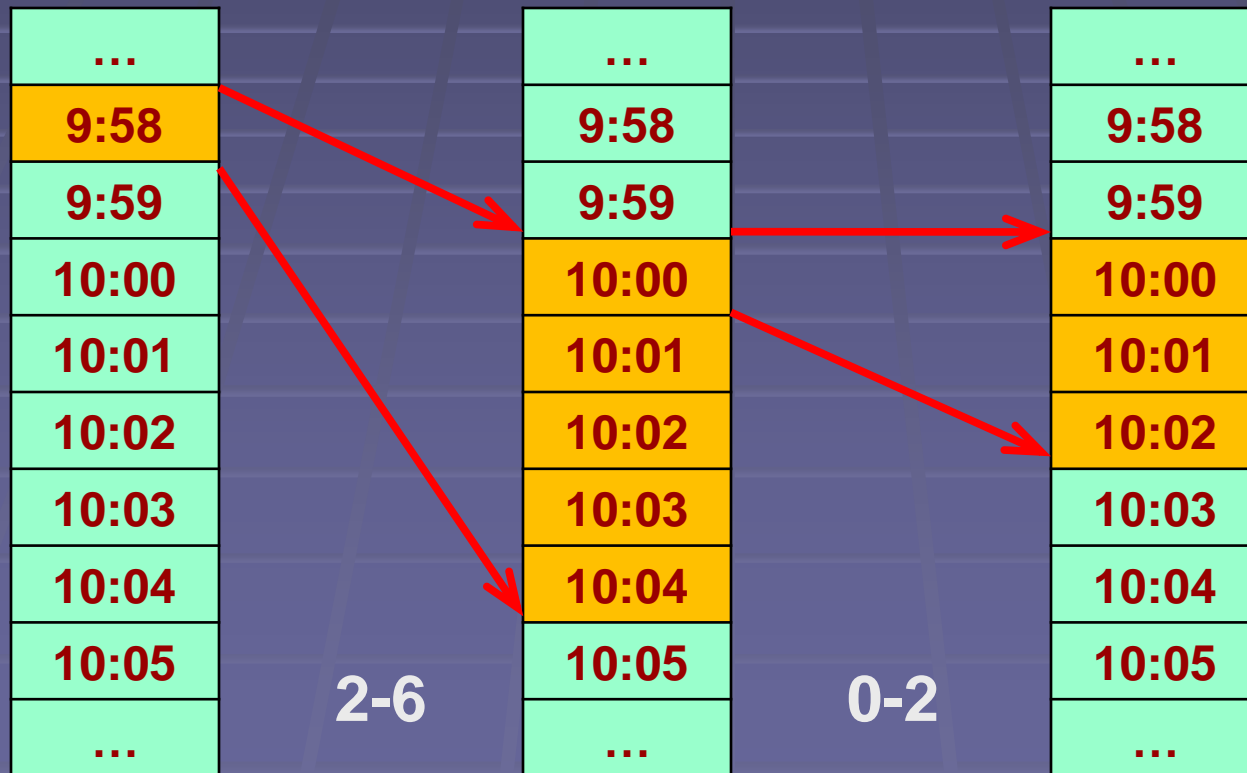
- ... bez komentáře
- 😊

# Clock



# Clock

- U každého vzorku zjistíme možné časy
  - To bylo už loni 😊





# Clock – řešení

- Pamatuju si, kolik předchozích a následujících vzorků je možných
- Postupně vyřazujeme, co možné není
- Možnost „vypadne“
  - → odečtu předchozí a následující
  - Kde se objeví nula, také není možné
- ... atd.

# Difficult



# Difficult – první nápad

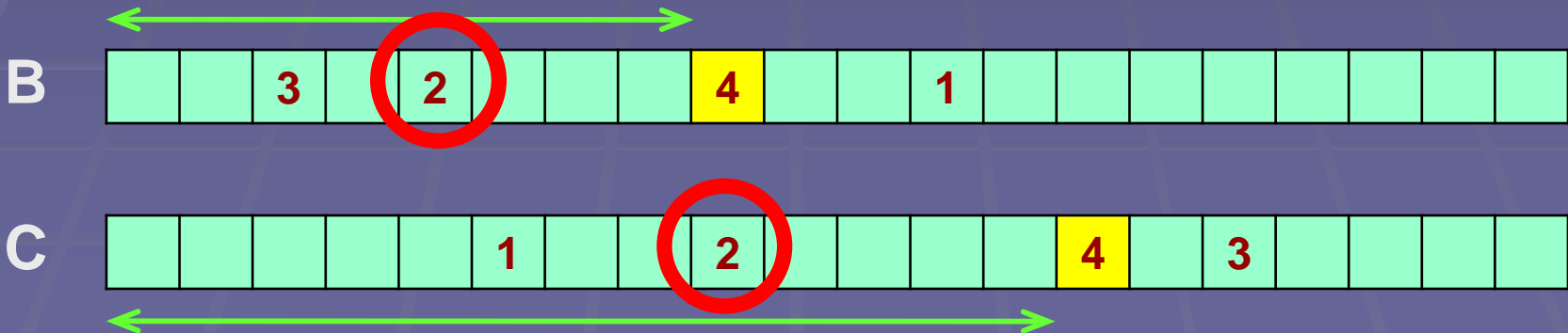
- Otestovat každé číslo s každým
  - Time limit exceeded ☹️

```
long long ret=0;
for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)
        if(ia[i]<ia[j] && ib[i]<ib[j] && ic[i]<ic[j])
            ret++;
```



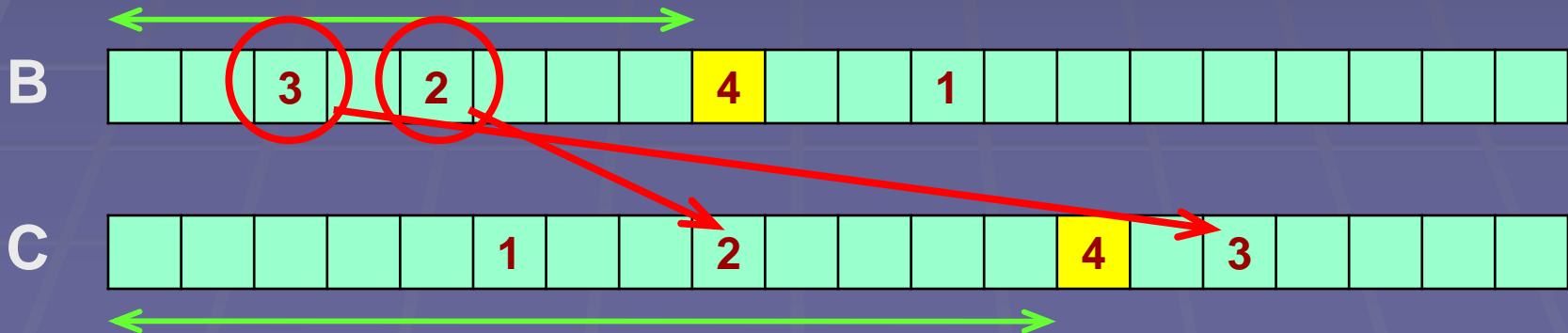
# Difficult – princip

- Při postupném přidávání
  - Přičteme všechna čísla, která
    - jsou menší
    - v obou posloupnostech jsou před novým
  - *No jo, ale jak?* 😊



# Difficult – princip

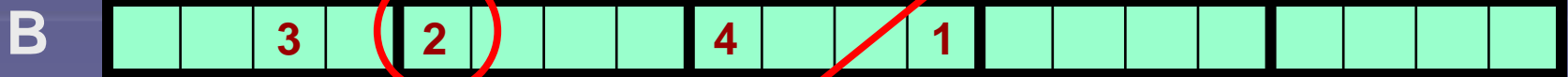
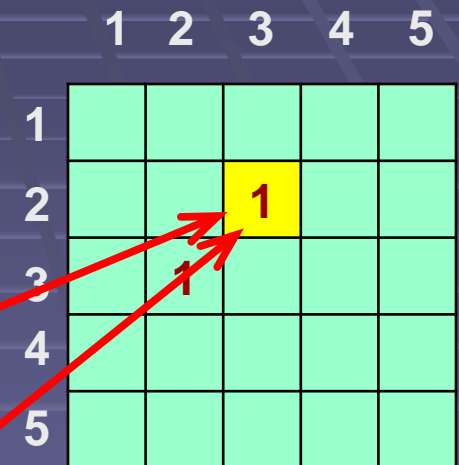
- Pro každé číslo z B zjistit, kde je v C
  - Kvadratická složitost ☹️





# Difficult – už to skoro je

- Rozdělíme pole na úseky
- Vždy, když přidáváme číslo, zapíšeme si, kde bylo



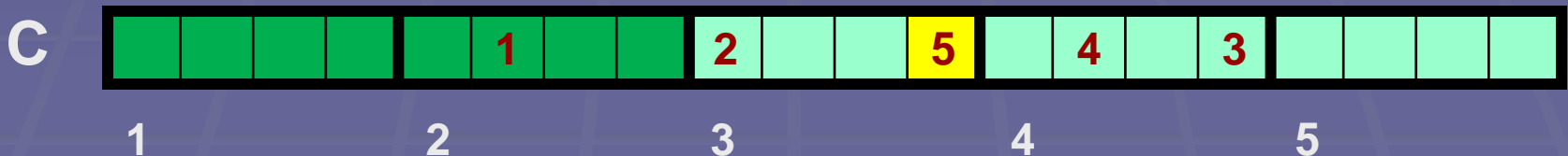




# Difficult – už to skoro je

- Můžeme pak zjistit všechny celé předchozí úseky (zelené)

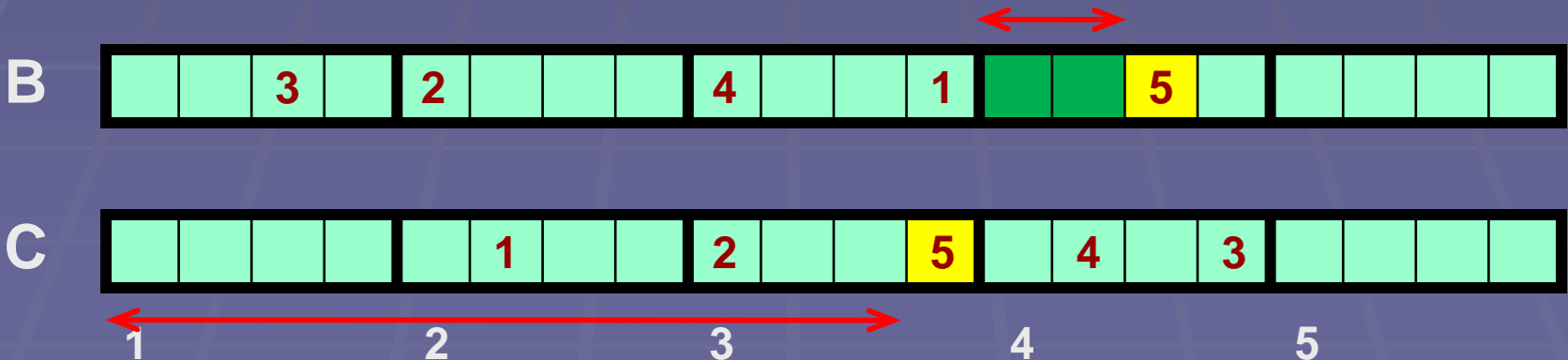
	1	2	3	4	5
1				1	
2			1		
3		1		1	
4					
5					



# Difficult – už to skoro je

- K tomu dopočítáme co chybí
  - Tj. co je v obou polích v místě červených šipek

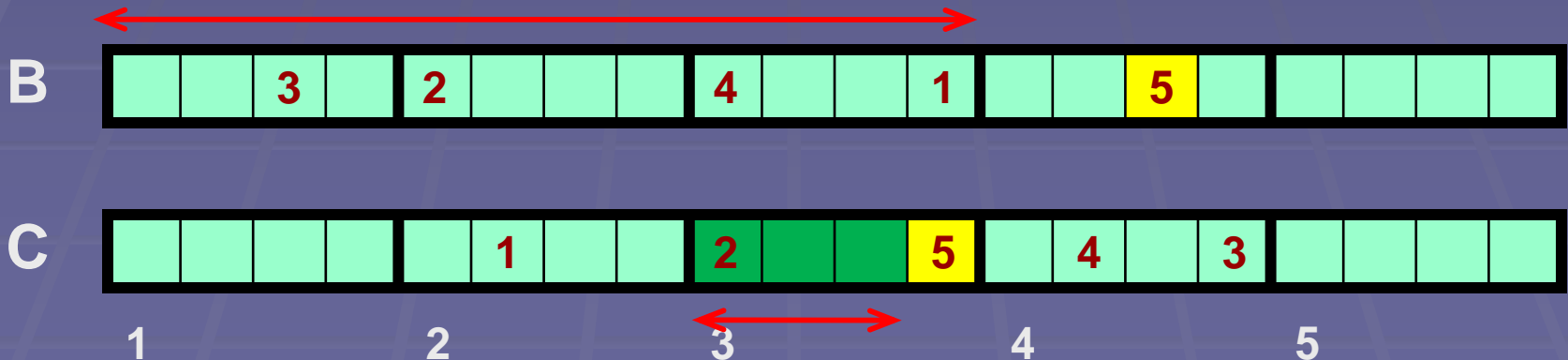
	1	2	3	4	5
1				1	
2		1			
3			1	1	
4					
5					



# Difficult – už to skoro je

- K tomu dopočítáme co chybí
  - Tj. co je v obou polích v místě červených šipek

	1	2	3	4	5
1				1	
2		1			
3			1	1	
4					
5					



# Difficult – složitost

- $n$  čísel,  $x$  velikost dílku
- $O((n/x)^2 + x)$ 
  - $\Rightarrow x = n^{2/3}$
  - $O(n^{2/3})$  jedno číslo
  - $\Rightarrow O(n^{5/3})$  celkem
  
- ... a nebo to šlo ještě jinak 😊

# Difficult – alternativní řešení

- Sečteme shody po dvojicích permutací
  - $A \times B$
  - $A \times C$
  - $B \times C$
  - celkem:  $S_3$
- To lze v čase  $O(n \cdot \log n)$ 
  - Binární stromy

# Difficult – alternativní řešení

- Pro každou dvojici čísel pak nastává právě jedno z následujících:
  - Všechny 3 permutace se na ní shodnou
  - 2 se shodnou a třetí je obráceně
- Celkem existuje  $N \cdot (N-1) / 2$  dvojic
  - Ty, které nás zajímají, se přičtou  $3x$
  - Ostatní jen  $2x$
- Výsledek:  $S3 - N \cdot (N-1)$

# Easy





# Easy – řešení

```
int digsum(int x) {
    int s = 0;
    while (x > 0) {
        s += x%10;
        x /= 10;
    }
    return s;
}

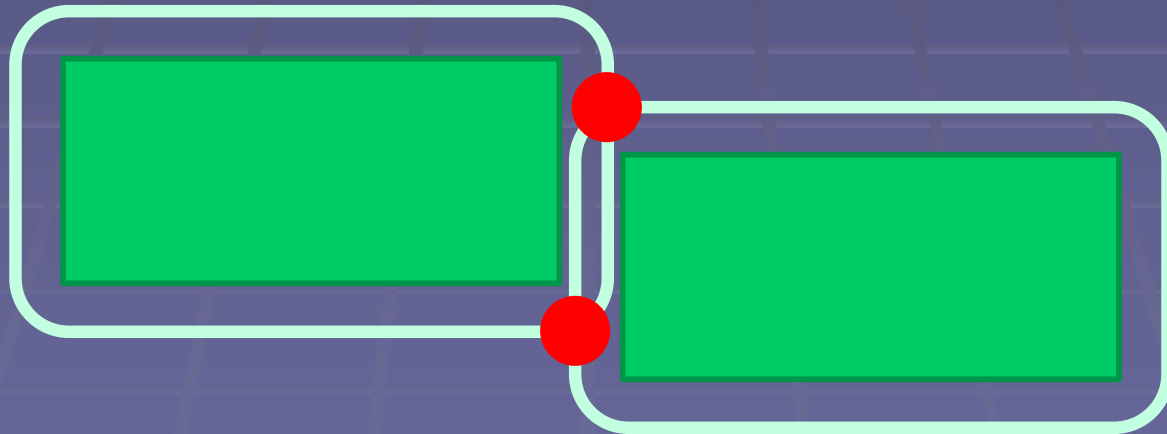
int sum = digsum(n);
while (sum = digsum(r * n))
    ++r;
```

# Fumar



# Fumar

- Řešení je na průsečících obvodů
  - Stačí taková místa otestovat
  - Pozor na časový limit



# Guards



# Guards – řešení

- Max 120 minut
  - Po méně než  $120^2$  se budou opakovat
  - Stačí sledovat pohyb stráží
- Pozor na reálná čísla
- Jak zjistit vzdálenost?



# Guards

```
double mindist2(double xa1, ya1, za1, xa2, ya2, za2,
                xb1, yb1, zb1, xb2, yb2, zb2) {
    double xd1 = xa1 - xb1, yd1 = ya1 - yb1, zd1 = za1 - zb1;
    double xd0 = xa2 - xb2 - xd1, yd0 = ya2 - yb2 - yd1, zd0 = za2 - zb2 - zd1;
    double nom = xd0*xd1 + yd0*yd1 + zd0*zd1;
    double den = sqr(xd0) + sqr(yd0) + sqr(zd0);
    double d1 = pyth2(xa1, ya1, za1, xb1, yb1, zb1);
    double d2 = pyth2(xa2, ya2, za2, xb2, yb2, zb2);
    if (d2 < d1) d1 = d2;
    if (den > EPS && nom < EPS && -nom < den) {
        double t = -nom / den;
        d2 = pyth2(xa1+t*(xa2-xa1), ya1+t*(ya2-ya1), za1+t*(za2-za1),
                  xb1+t*(xb2-xb1), yb1+t*(yb2-yb1), zb1+t*(zb2-zb1));
        if (d2 < d1) d1 = d2;
    }
    return d1;
}
```

# Hack the Lock





# Hack – 1 číslice

- $X \leq 4$

- $X(X-1) \dots 10123456789$

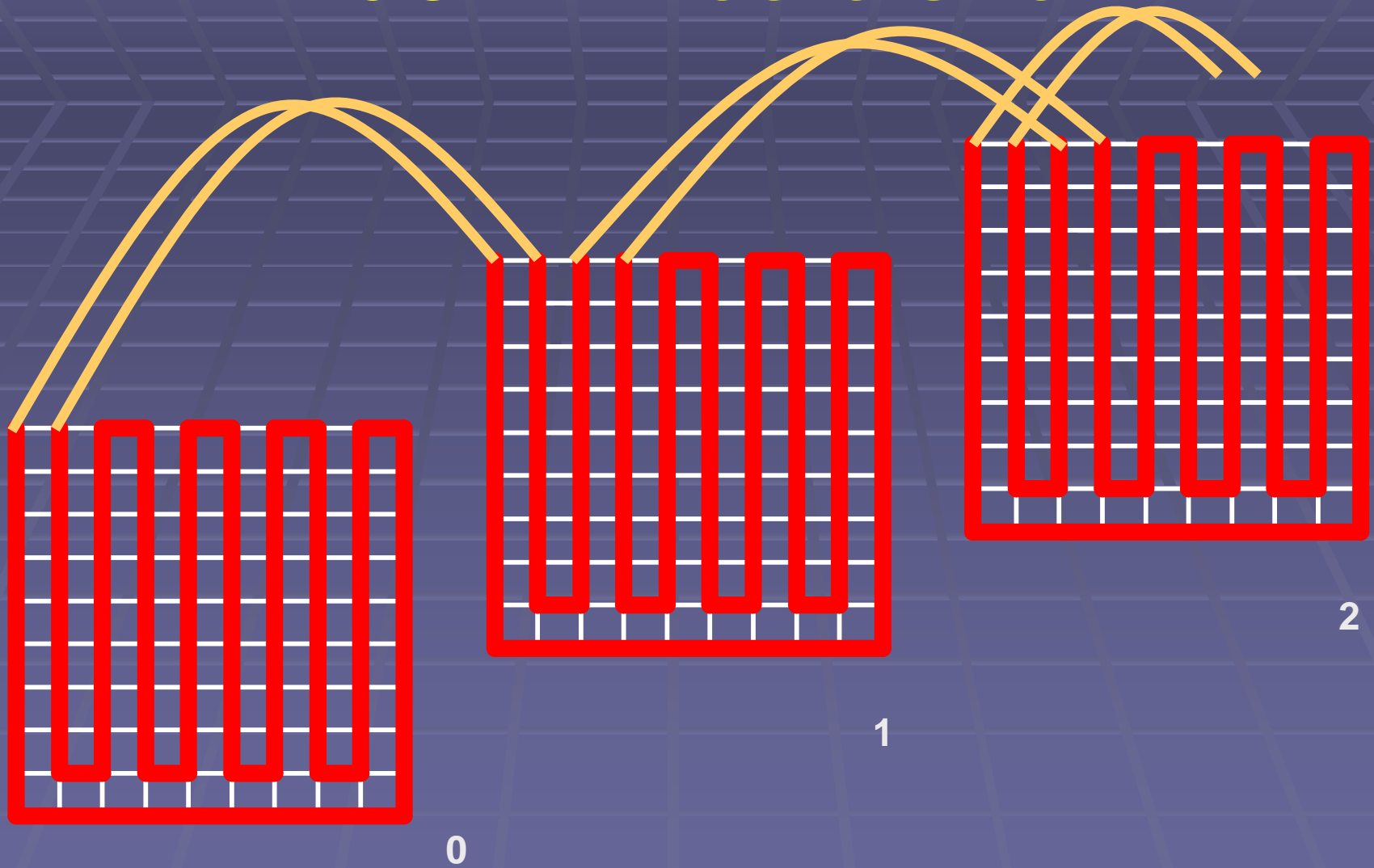
- $X \geq 5$

- $X(X+1) \dots 789876543210$





# Hack – více číslic



# The $l$ -th queen











# Queens – kód

```
usedx = new int[x+1]; usedy = new int[y+1];
useds = new int[x+y+1]; usedd = new int[x+y];
while (d-- > 0) {
    int xi = nextInt(), yi = nextInt();
    ++usedx[xi]; ++usedy[yi];
    ++useds[xi+yi]; ++usedd[y+xi-yi];
}

for (int i = 1; i <= x; ++i)
    for (int j = 1; j <= y; ++j)
        if (usedx[i] == 0 && usedy[j] == 0
            && useds[i+j] == 0 && usedd[y+i-j] == 0)
            ++cnt;
```

# Autoři úloh

**Josef Cibulka**

**Zdeněk Dvořák**

**Martin Kačer**

**Jan Stoklasa**

**Jakub Černý**

**Jan Katrenic**

**Radek Pelánek**