

Sample Solutions



CTU Open Contest 2013

Czech Technical University in Prague

FRACTIONAL LOTION

FL – úloha

- $1/n = 1/x + 1/y$
- Vyzkoušíme všechny možnosti
 - Jaké je omezení na x a y ?
 - Buď $1/x \geq 1/n/2$ nebo $1/y \geq 1/n/2$
 - \Rightarrow Jedno z nich je maximálně $2 \cdot n$

FL – řešení

```
int solve(int n) {  
    int c = 0;  
    for (int a = n+1; a <= n*2; ++a)  
    {  
        if (a * n % (a-n) == 0)  
            ++c;  
    }  
    return c;  
}
```

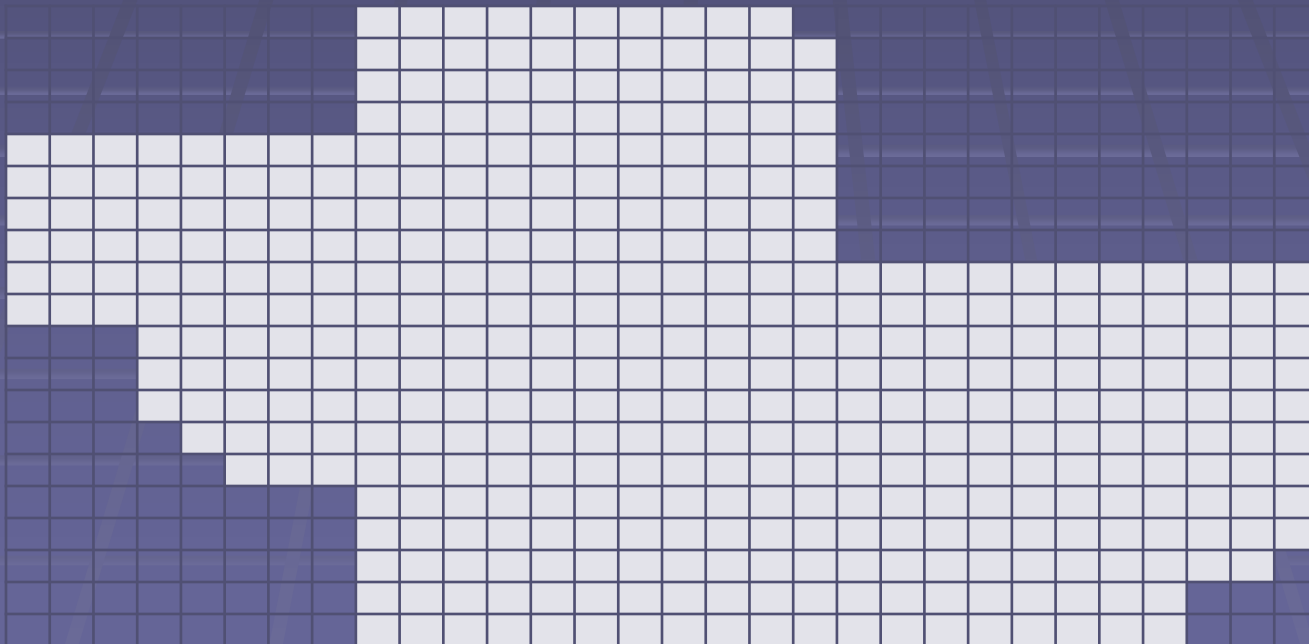
FL – poznámky

- Lze řešit rychleji, ale nebylo požadováno
- (mj. na to naváděla poznámka na úvodní straně)

FOLDED MAP

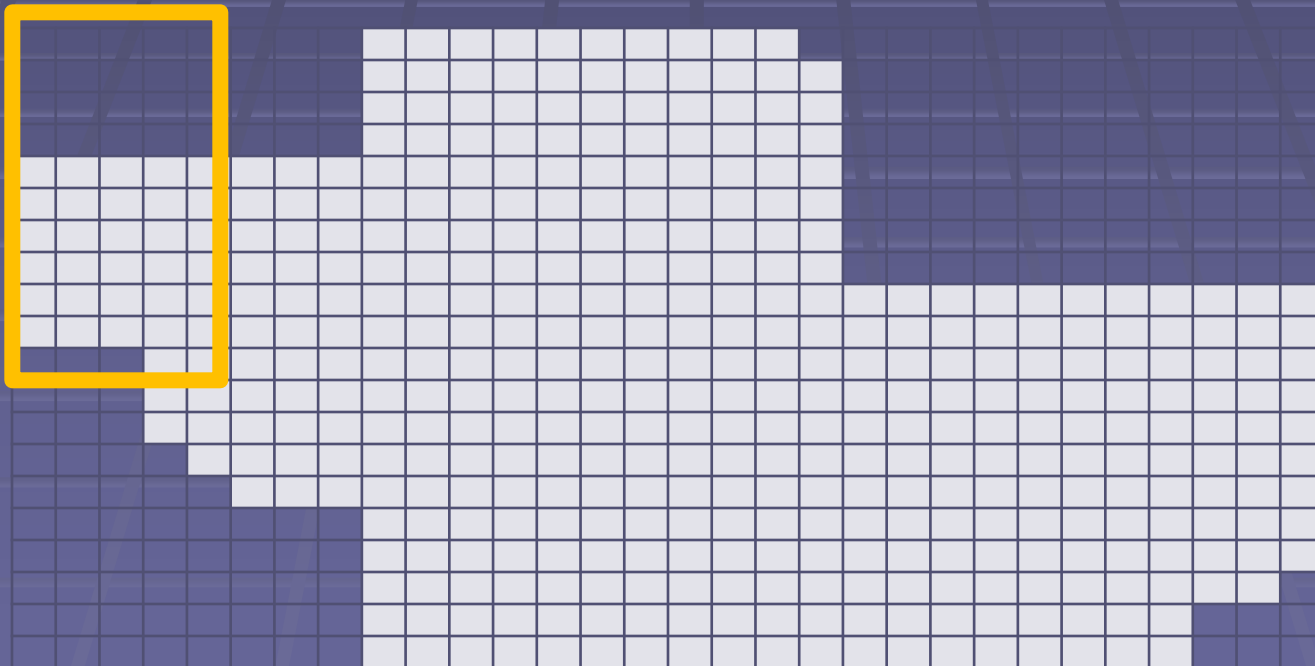
FM – úloha

- Základní myšlenka
 - Vyzkoušíme všechny možnosti posunu



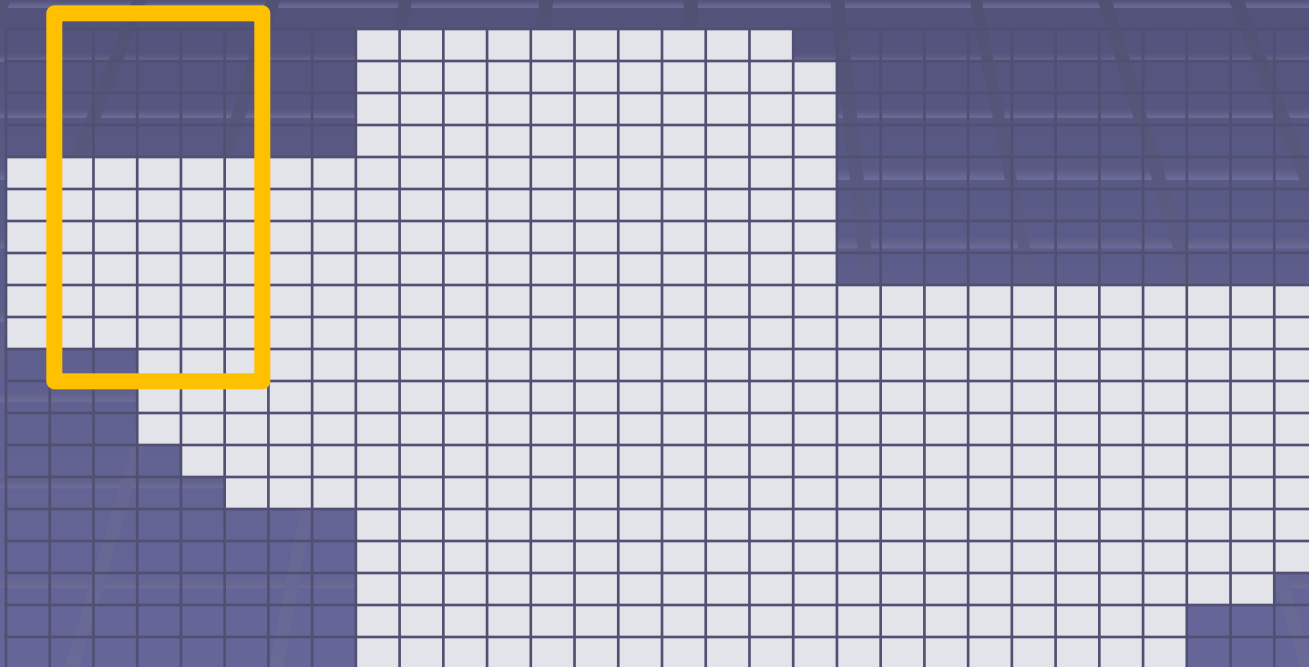
FM – úloha

- Základní myšlenka
 - Vyzkoušíme všechny možnosti posunu



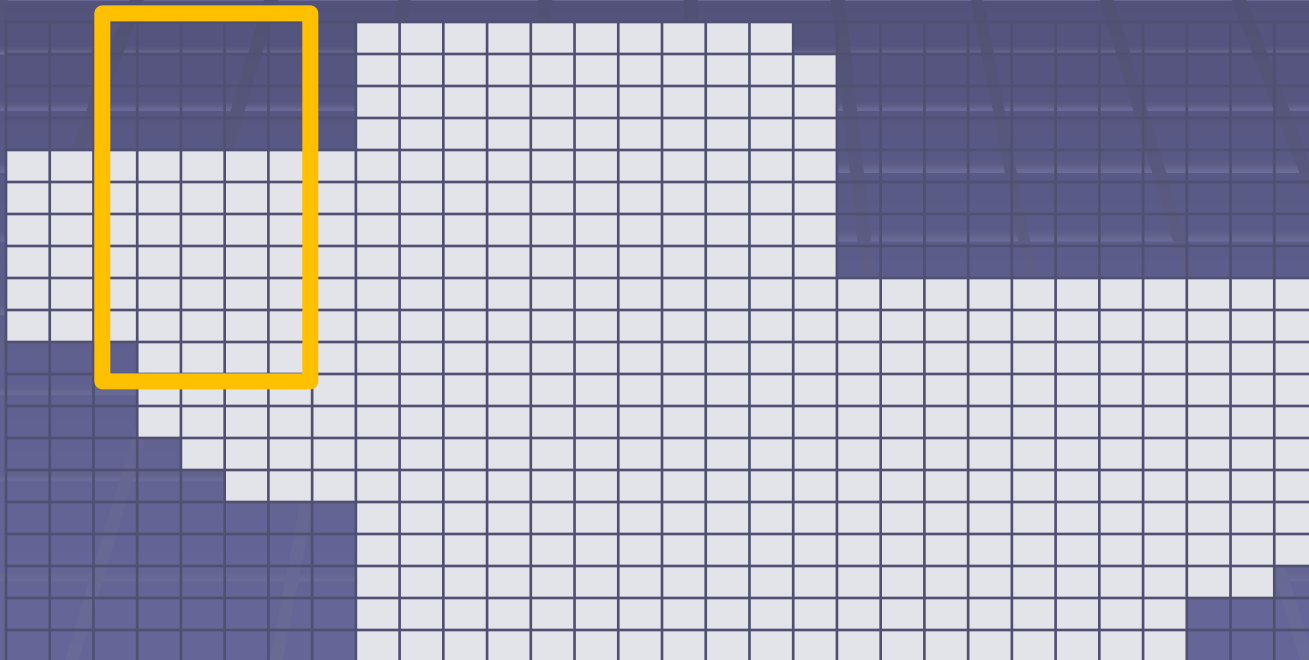
FM – úloha

- Základní myšlenka
 - Vyzkoušíme všechny možnosti posunu



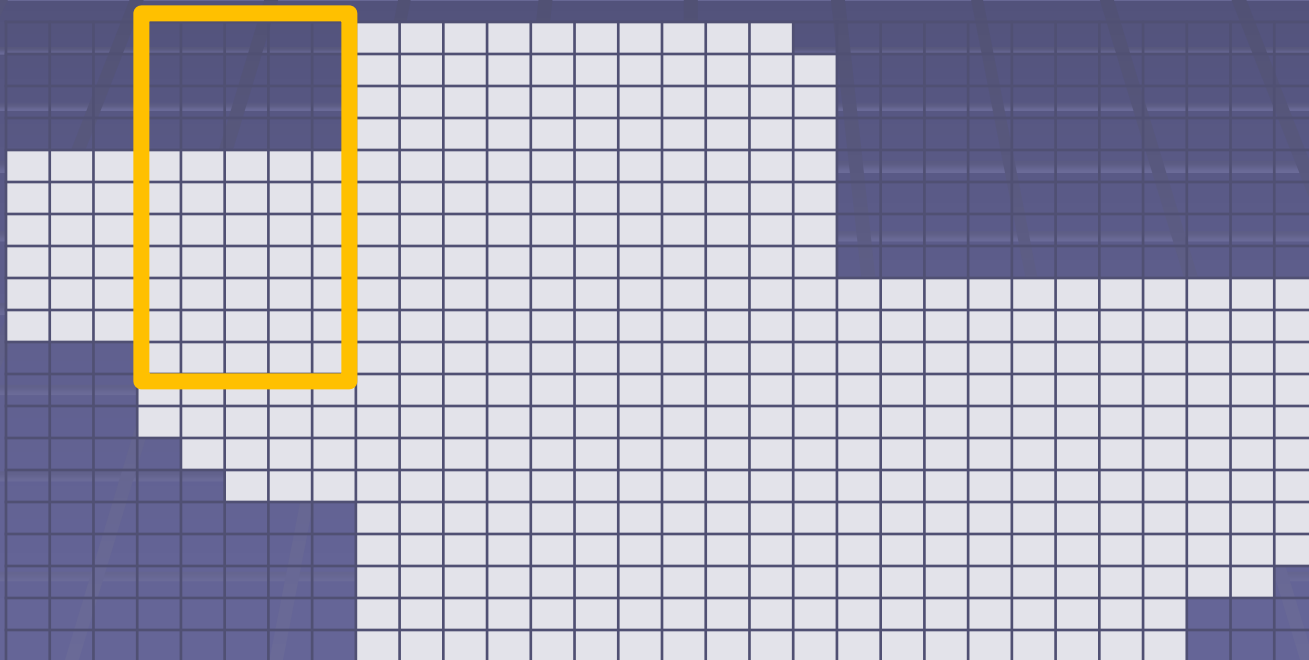
FM – úloha

- Základní myšlenka
 - Vyzkoušíme všechny možnosti posunu



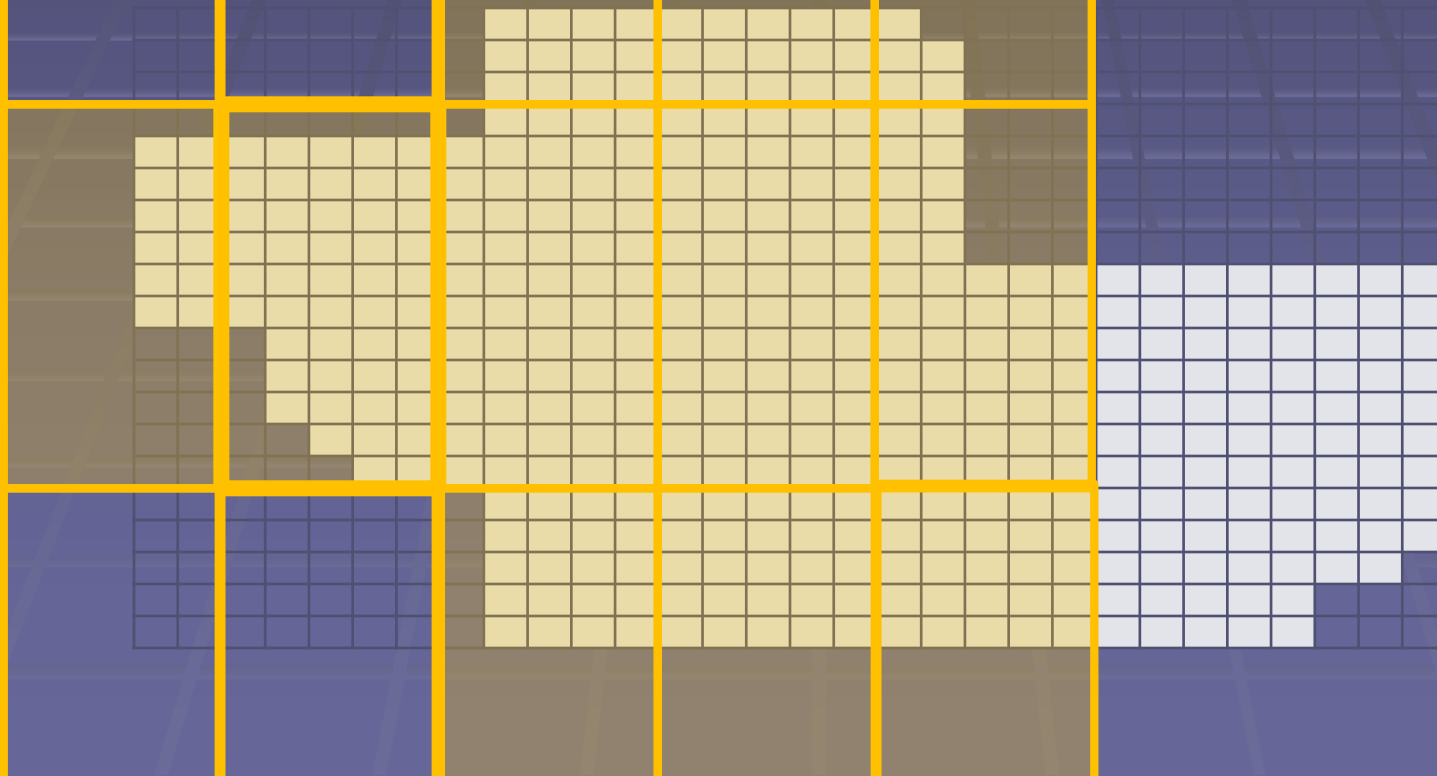
FM – úloha

- Základní myšlenka
 - Vyzkoušíme všechny možnosti posunu



FM – řešení

- Pro každý posun spočítám počet potřebných stránek



FM – algoritmus

- Jaká je operační složitost
 - Počet posunutí: $px \times py$
 - Počet stránek: $(mx/px+1) \times (my/py+1)$
 - \Rightarrow Celkově $mx \times my$ testů (lineárně)

- Jeden test chceme efektivně

FM – efektivita

- Jak testovat zaplnění stránky?
 - Předpočítat na začátku volné pixely (světle modrá značí volno)

1	2	3	4	5	0	0	1	2	3	4	5
1	2	3	4	5	0	0	1	2	3	4	0
1	2	3	4	0	1	0	1	2	3	4	5
1	2	3	4	0	1	2	0	1	0	1	2

FM – efektivita

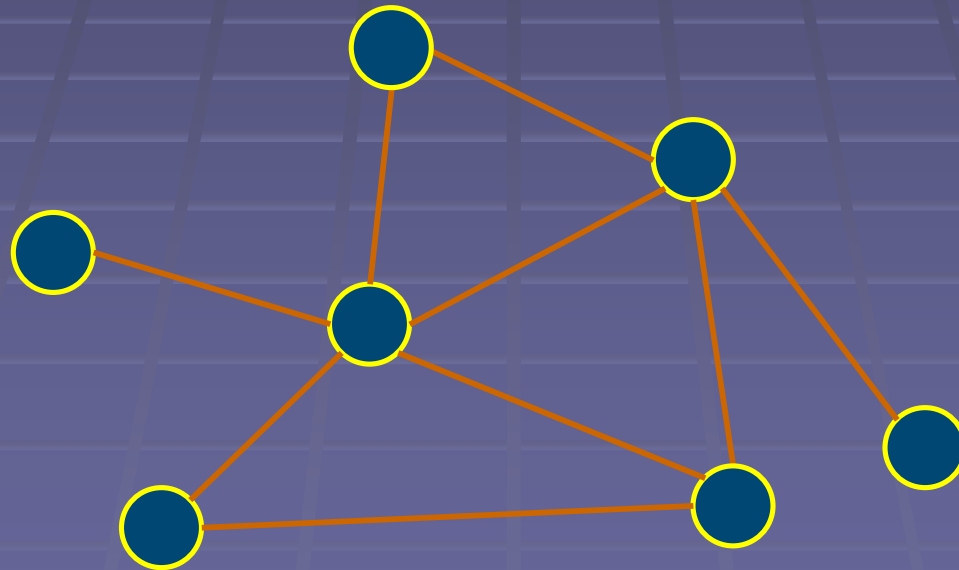
- Předpočítání volných stránek
 - Nyní počítám „zelená“ směrem dolů

0	0	0	1	1	0	0	0	0	0	1	1
0	0	0	2	2	0	0	0	0	0	2	0
0	0	0	3	0	0	0	0	0	0	3	1
0	0	0	4	0	0	0	0	0	0	0	0

FURRY NUISANCE

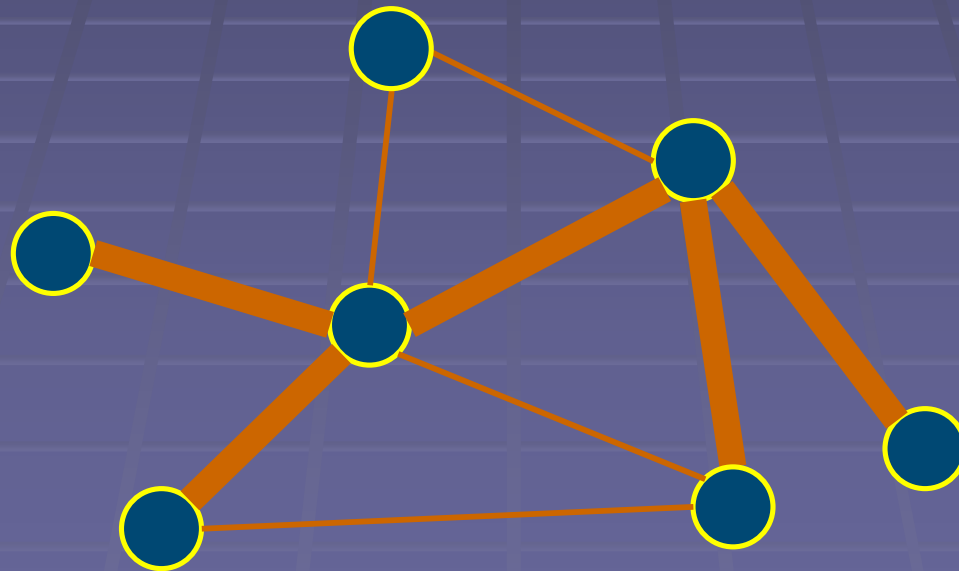
FN – rekapitulace zadání

- Hledáme kostru se 4 listy
 - (původně mělo být číslo vyšší)



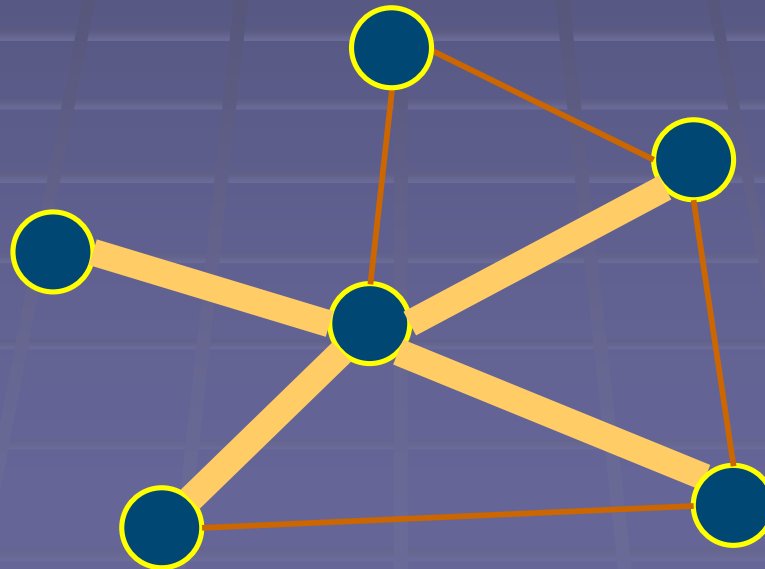
FN – rekapitulace zadání

- Hledáme kostru se 4 listy
 - (původně mělo být číslo vyšší)



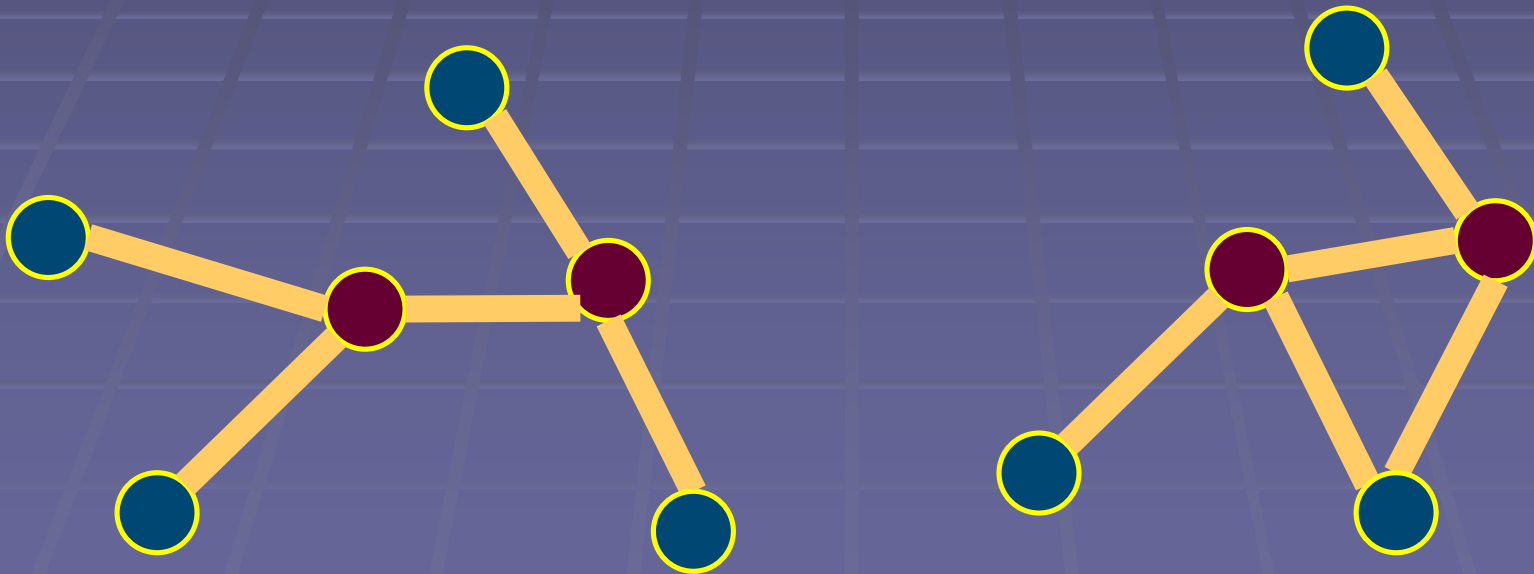
FN – princip

- Co způsobí kostru se 4 listy?
 - Uzel stupně 4 (a více)
 - Nebo 2 uzly stupně 3 (ve stejné komp.)



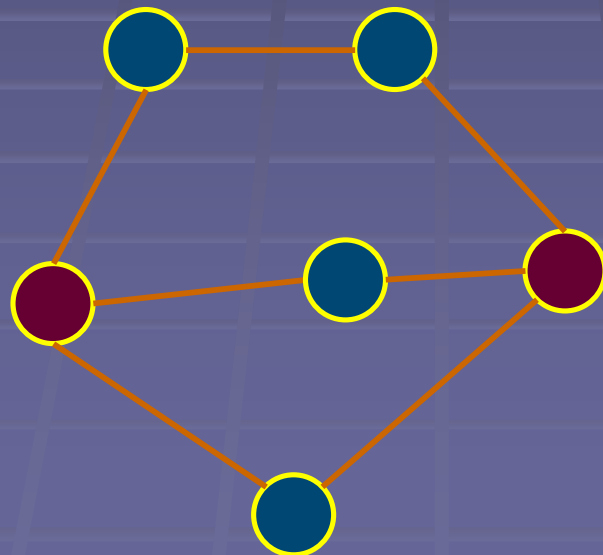
FN – speciální případy

- Ale pozor!
 - Mohou spolu sdílet sousedy...



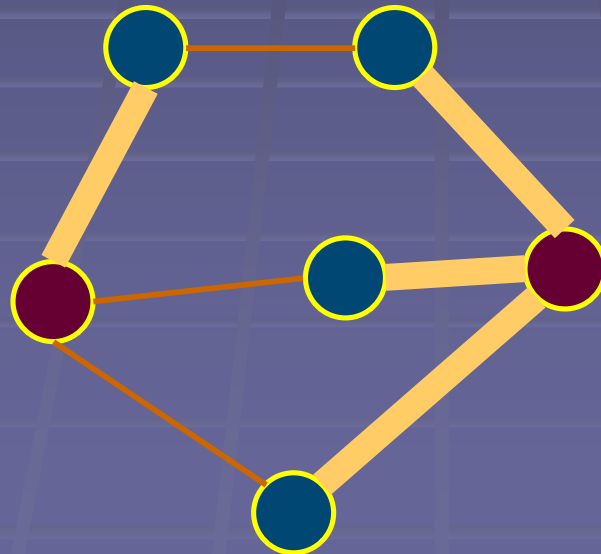
FN – pro uzly stupně 3

- Ale pozor!
 - Dokonce ani nestačí, když mají 4 susedy dohromady



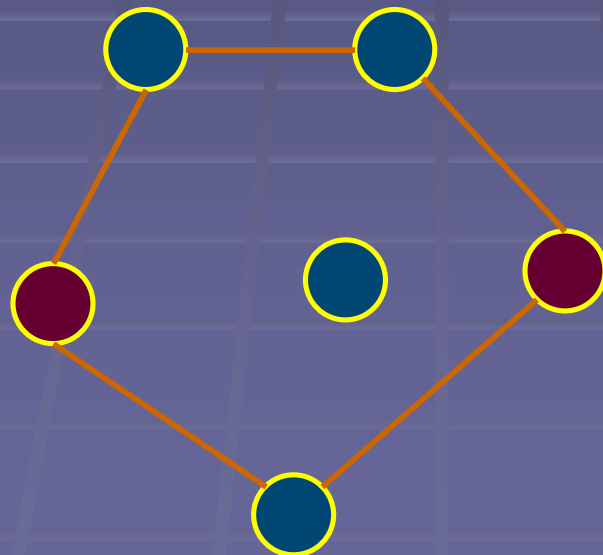
FN – pro uzly stupně 3

- Ale pozor!
 - Dokonce ani nestačí, když mají 4 susedy dohromady



FN – pro uzly stupně 3

- Nejkratší cestu mezi nimi odeberu
- Musí ještě zůstat alespoň 4 sousedé



FN – algoritmus

- Lze vyzkoušet pro všechny dvojice uzlů stupně 3 (nemohou být všechny u sebe)

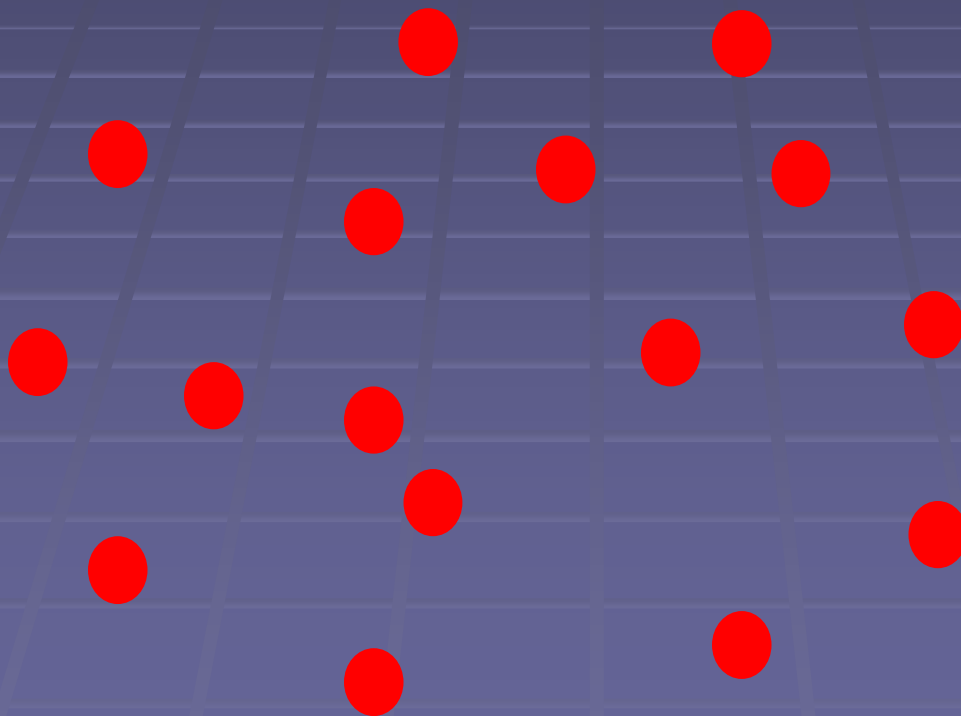
FENCE ORTHOGONALITY

FO – zadání

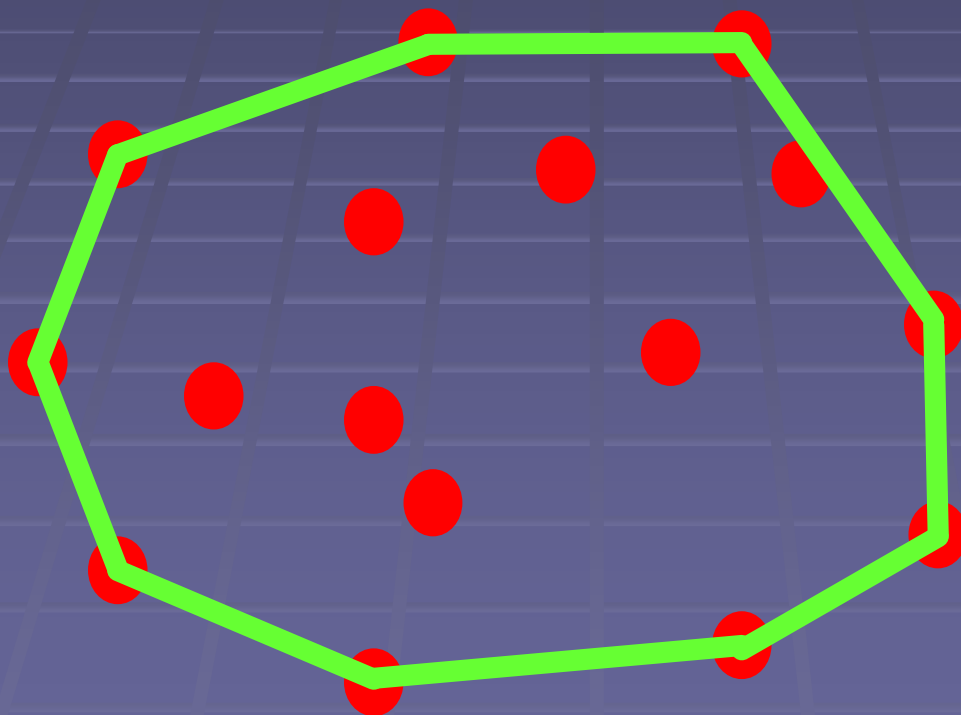
- Obdélník kolem daných bodů
- Nejmenší obvod

- „Obdélníková konvexní obálka“

FO – konvexní obálka

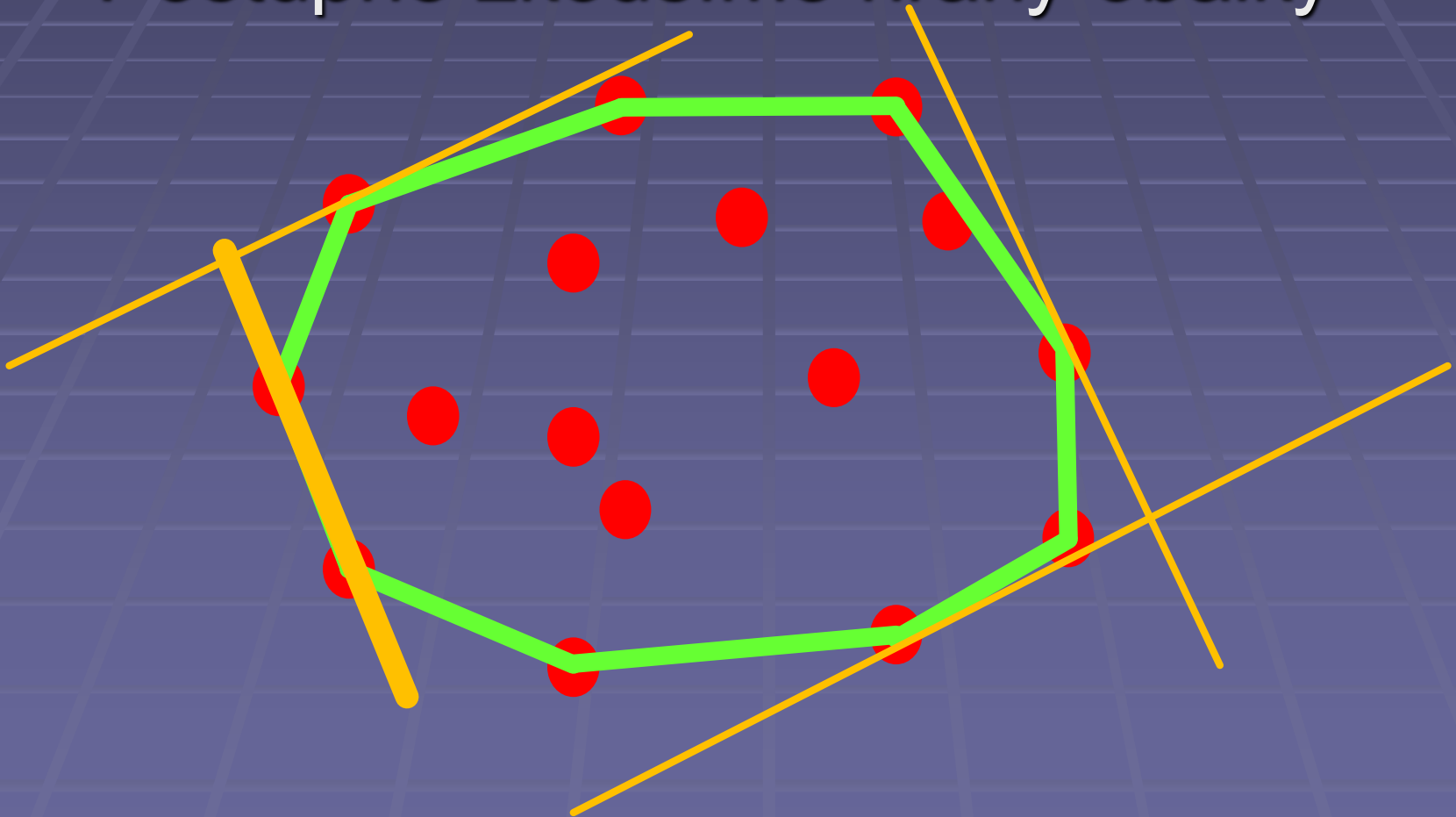


FO – konvexní obálka



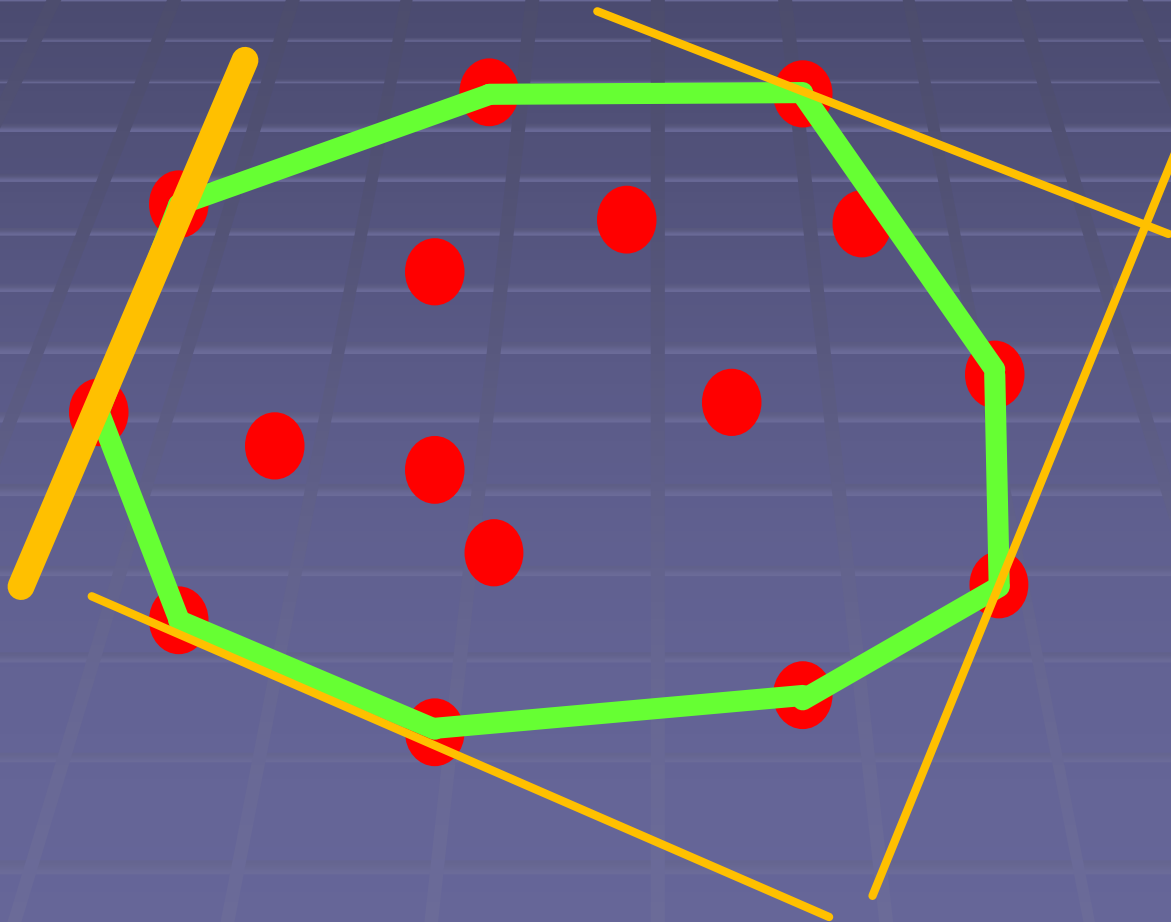
FO – konvexní obálka

- Postupně zkoušíme hrany obálky

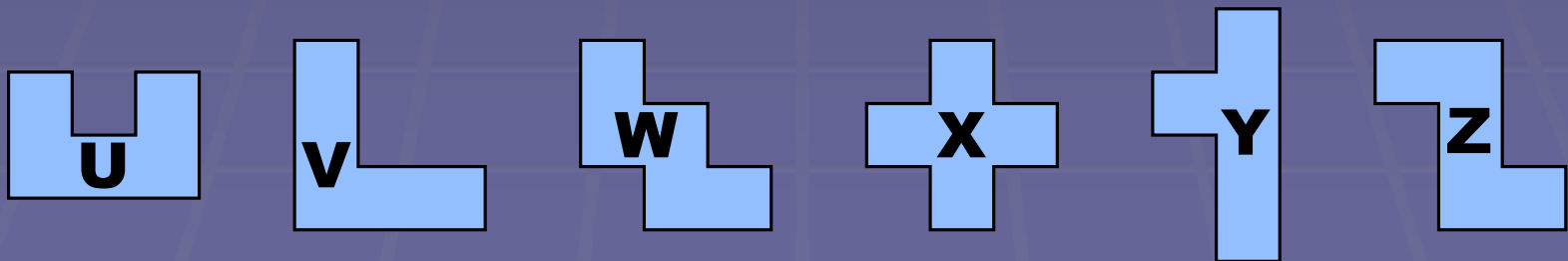
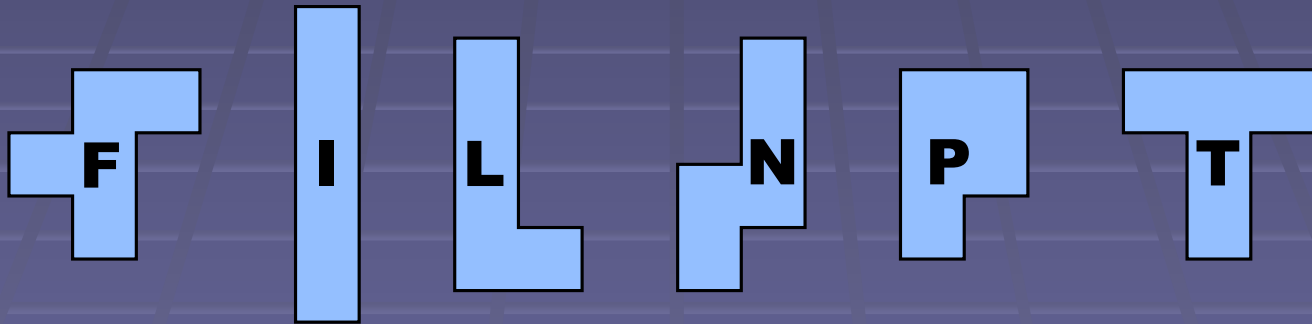


FO – konvexní obálka

- Lze lineárně (postupně „otáčet“)

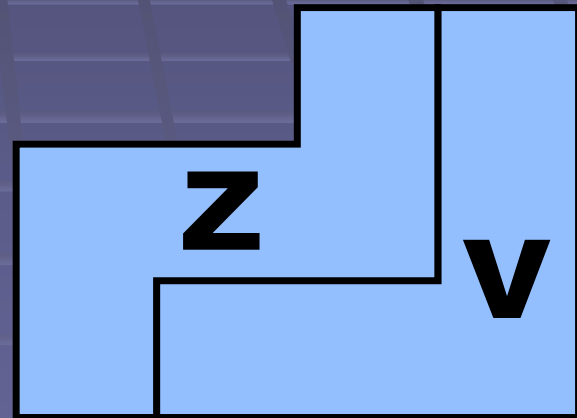
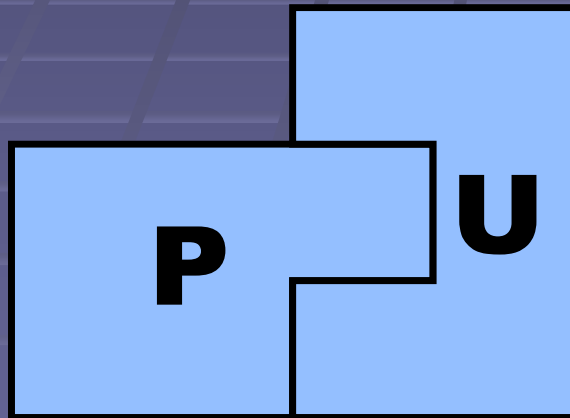


FLOWER POTS



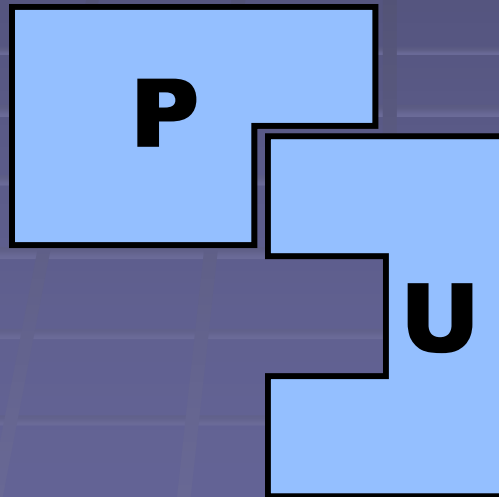
FP – když to nejde silou...

- Vyzkoušíme všechny možnosti



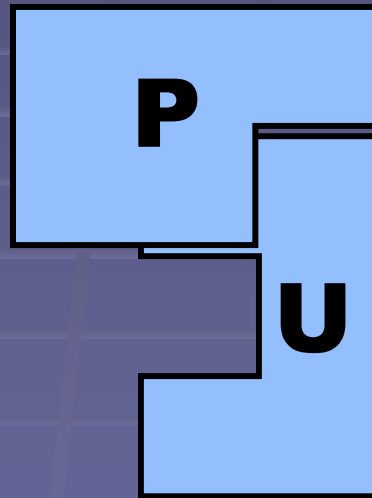
FP – když to nejde silou...

- Různá umístění...



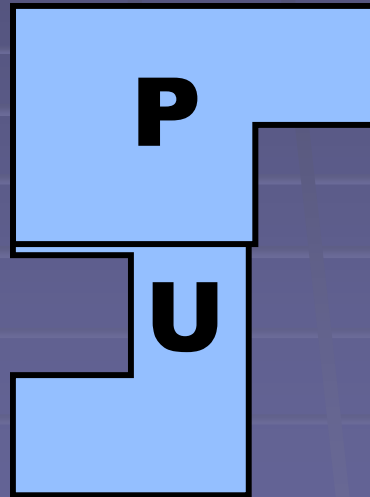
FP – když to nejde silou...

- Různá umístění...



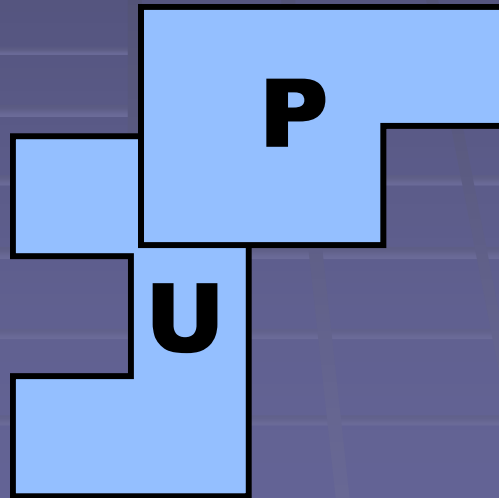
FP – když to nejde silou...

- Různá umístění...



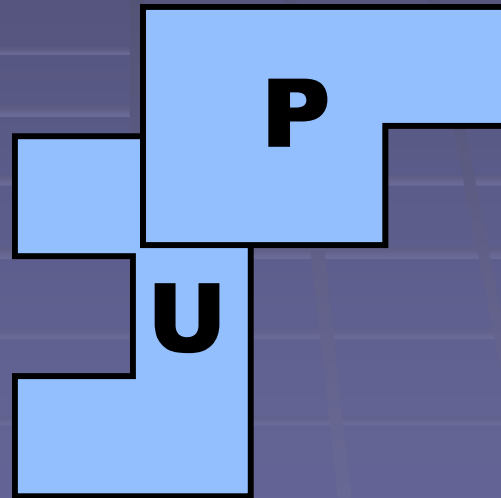
FP – když to nejde silou...

- Různá umístění...



FP – když to nejde silou...

- Různá umístění...
- ... plus rotace a otáčení



FRUSTRATED QUEUE

FQ – zadání

- Doplnění závorek na „platný“ zápis
- Dynamické programování
- Udržujeme v poli, kolika způsoby lze dosáhnout daného počtu otevíracích závorek

FQ – zadání

- Na začátku jen 1 způsob pro 0 levých

0	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0

FQ – zadání

- Přejde levá závorka => posunu vpravo

((



0	1	2	3	4	5	6	7	8
0	1	0	0	0	0	0	0	0

FQ – zadání

- Přejde tečka => sečtu vlevo i vpravo

((.



0	1	2	3	4	5	6	7	8
0	0	1	0	0	0	0	0	0

FQ – zadání

- Přejde tečka => sečtu vlevo i vpravo

((..



0	1	2	3	4	5	6	7	8
0	1	0	1	0	0	0	0	0

FQ – zadání

- Přejde tečka => sečtu vlevo i vpravo

((..

0	1	2	3	4	5	6	7	8
1	0	2	0	1	0	0	0	0

FQ – zadání

- Pravá závorka => posunu doleva

((..))



0	1	2	3	4	5	6	7	8
1	0	2	0	1	0	0	0	0

FQ – zadání

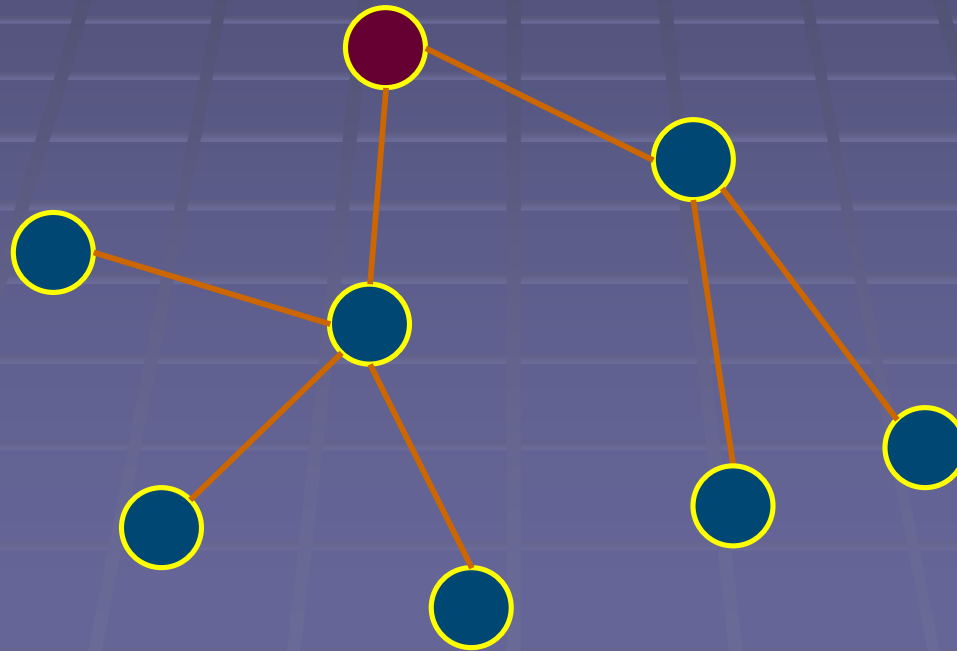
- Na konci výsledek v prvním prvku

0	1	2	3	4	5	6	7	8
1	0	2	0	1	0	0	0	0

FROZEN ROSE HEADS

FR – úloha

- Nejlevnější způsob jako „odříznout“ listy stromu (kořenového)



FR – úloha

- Rekurze (1000 uzlů)
 - Sečtu všechny hrany „dolů“ (rekurzivně)
 - Porovnáám s hranou „nahoru“
- Výsledek vrátím

FR – program

```
int dfs(int u, int parent, int w) {
    int i, chld = 0, leaf = 1;
    for (i = 0; i < nc[u]; ++i) {
        if (nu[u][i] != parent) {
            leaf = 0;
            chld += dfs(nu[u][i], u,
                       ew[u][nu[u][i]]);
        }
    }
    return (w < chld || leaf) ? w : chld;
}
```

FALSE SENSE OF SECURITY

FS – přímočaré řešení

```
for (char c : s.toCharArray()) {
    String morch = char2string.get(c);
    sizes[cnt++] = morch.length();
    encoded.append(morch);
}
int pos = 0;
while (--cnt >= 0) {
    int b = pos + sizes[cnt];
    System.out.print(string2char.get(
        encoded.substring(pos, b)));
    pos = b;
}
```

DOTAZY



Autoři úloh

Josef Cibulka

Zdeněk Dvořák

Marko Genyk-Berezovskyj

Martin Kačer

Jan Stoklasa

Tradičně náměty: **Radek Pelánek**

Greater NY & Stanford Local Contest