

Bude podobný u tabule: Všechní Goldberger alg., jinak se sumštem přičk.

Goldberger algoritmus

- 1) $\forall v: h(v) = 0, h(2) = n$
- 2) $\forall c: f(c) = 0$
 $\forall 2v \in E: f(2v) = c(2v)$
- 3) Pokud $\exists u \neq 2v, f(u) > 0$
- 4) Pokud $\exists uv: r(uv) > 0, h(u) > h(v)$
- 5) Převést po uv.
- 6) Jinak:
- 7) $h(u) = h(u) + 1$

1) Jednoduchý Goldberg

- Jak se bude chovat pro jednoduché případy? Bude rychlejší?

Moje hypotéza je, že se radikálně sníží počet převodů:

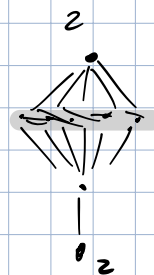
Celkem nenasycených převodů:

Pro jednoduché případy neexistuje nenasycený převod, protože jakmile převod, tak jsem musel přerůst vše (to jedinou 1).

Celkem tam bude $m \cdot n$ převodů a nenasycených převodů.

Odhad skom
- existují pouze
násazení
v $O(n \cdot m)$

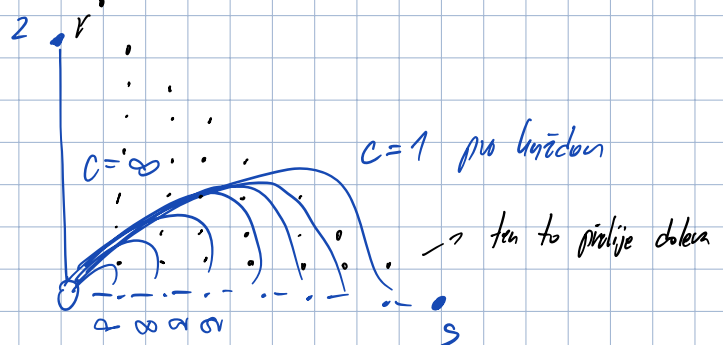
Odhad zobra:



→ tabulky to nalejete do
jednotky vektoru, ale pak začnete
zvětšovat ty další

tabulky špatně vyglejeme do zdijs

2



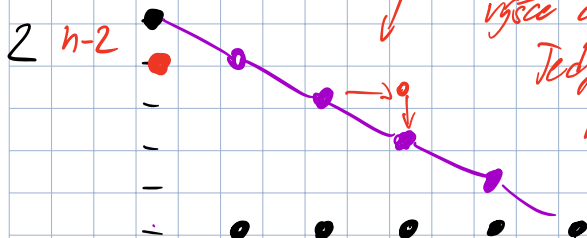
2) Nízky zdroj

$n-1$

Takže v nejhorším
stavu má stejné
výše co zdroj:
Tedy nic
neutratit.

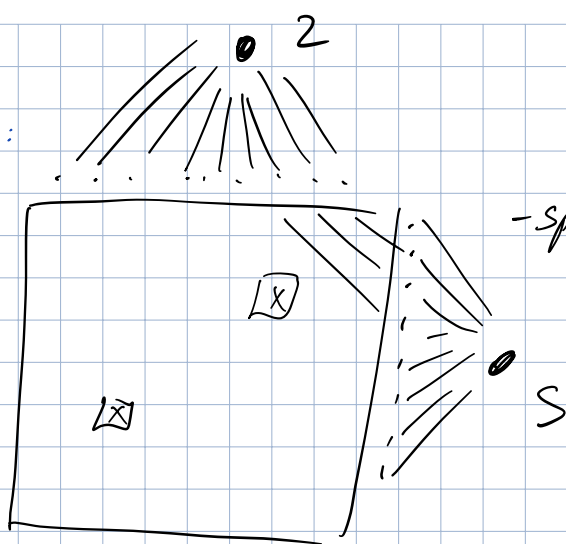
$n-3$

$n-2$



S

3) Věze:



- spojíme každý vrchol, v jejichž pravidelných
vertikál a horizontál není díra.

Pak hledáme tak, ten zajistí největší plování!

> Je potřeba pouze zajistit, aby byly bezpečné
hmny jednotkové.

4) Věze II Plund je sloupec / řádek rozdělen dírami, reprezentující ho samostatnými
vrcholy. Princip zůstává stejný.