Linear Programming

vector of unknowns

$$\underbrace{\begin{bmatrix} A \end{bmatrix}}_{n} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} \leftarrow \text{RHS}$$

linear system of equations

$$\underbrace{\begin{bmatrix} A \end{bmatrix}}_{n} \begin{bmatrix} x \end{bmatrix} \geq \begin{bmatrix} B \end{bmatrix}$$

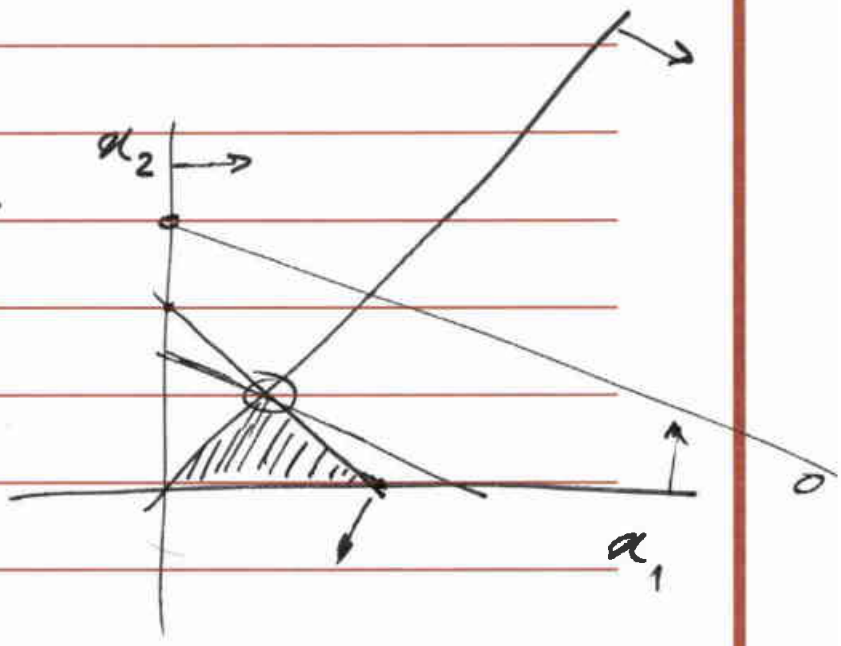
Objective function

$$\begin{bmatrix} c^T \end{bmatrix} \begin{bmatrix} x \end{bmatrix}$$

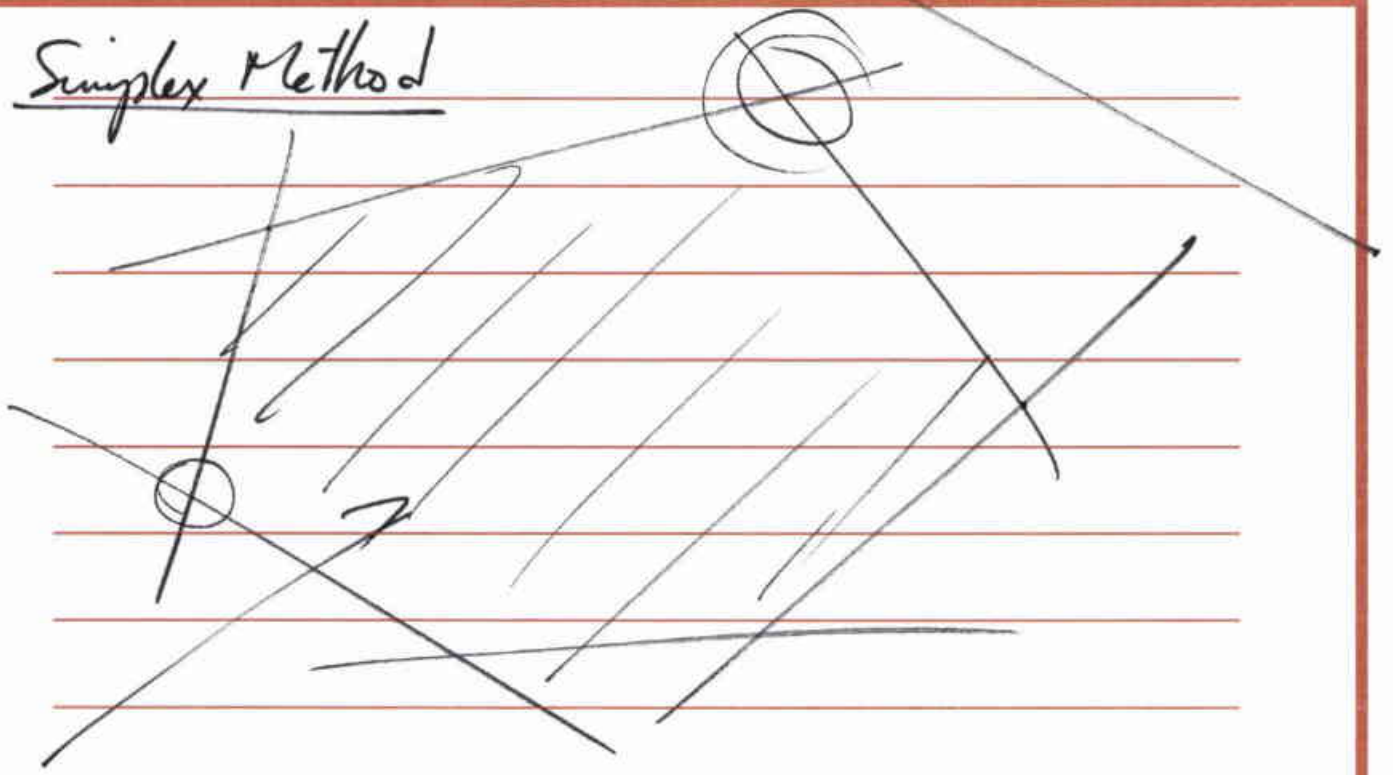
Minimize the objective function

$$\begin{cases} x_1 - x_2 \geq 0 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_1 + x_2 \leq 4 \end{cases}$$

Maximize ~~Minimize~~ $x_1 + 2x_2$



Simplex Method



Convex polygon

Weighted vertex Cover Problem

for $G=(V,E)$, $S \subseteq V$ is a set such that each edge has at least one end in S

$w_i \geq 0$ for each $i \in V$

$$W(S) = \sum_{i \in S} w_i$$

objective: Minimize $W(S)$

x_i is a decision variable for each node $i \in V$

$$\begin{cases} x_i = 0 & i \notin S \\ x_i = 1 & i \in S \end{cases}$$

$x_i + x_j \geq 1$ $\quad$ x_i $\quad$ x_j

Minimize $\sum w_i x_i$

Subject to

$$x_i + x_j \geq 1 \quad \text{for } (i,j) \in E$$

$$x_i \in \{0,1\} \qquad i \in V$$

Integer program

- Linear Programming — Continuous Variables
- integer " — discrete "
- Mixed integ. — both cont. & disc.
- non-linear "

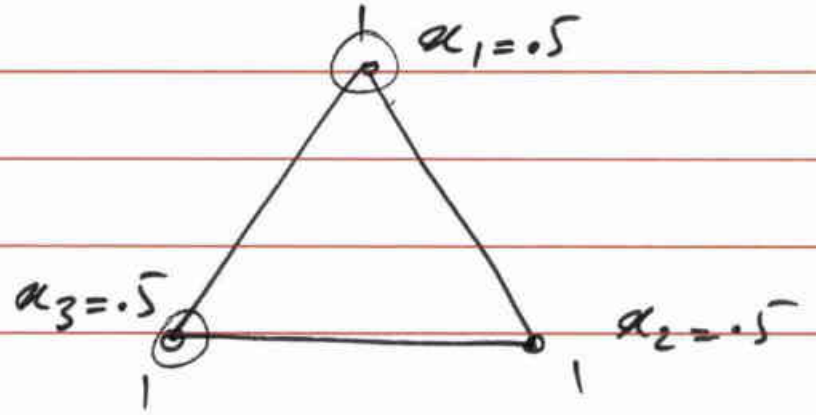
Drop the requirement That $x_i \in \{0,1\}$

and solve the LP in polynomial time and find $\{x_i^*\}$ between 0 & 1

$$W_{LP} = \sum_i w_i x_i^*$$

S^* is the opt. vertex cover set

$W(S^*)$ weight of the opt. sol.



$$W_{LP} = \sum w_i x_i^* = 1*.5 + 1.*.5 + 1.*.5$$
$$= 1.5$$

$W(S^*) = 2$

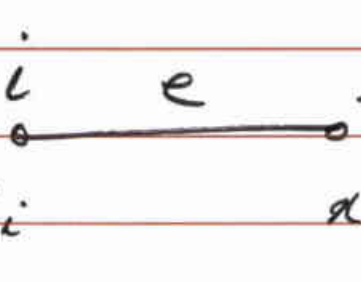
$W(S) = 3$

$x_i^* = 0 \Rightarrow i \notin S$

$x_i^* = 1 \Rightarrow i \in S$

Say $S_{\geq 1/2} = \{ i \in V : x_i^* \geq 1/2 \}$

$x_i + x_j \geq 1$



S is our approx. sol.

$$\begin{cases} w(S) \leq 2 * w_{LP} \\ w_{LP} \leq w(S^*) \end{cases}$$

$$w(S) \leq 2 * w(S^*)$$

a 2-approximation

Max. Flow Problem

variables are flows over edges

objective function: Maximize $\sum_{e \text{ out of } s} f(e)$

Subject to $0 \leq f(e) \leq c_e$ for each edge $e \in E$

$\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) = 0$ for $v \in V$

except for s & t.

$A = B$

$\begin{cases} A \leq B \\ B \leq A \end{cases}$

Multi commodity flow:

$f_i(e)$: flow of commodity i over edge e.

α_i : is the profit associated w/ one unit of flow for commodity i.

we have m commodities

Objective: Maximize profit

$$\text{Maximize} \sum_{i=1}^{m} \sum_{\text{edges}} \alpha_i f_i(e)$$

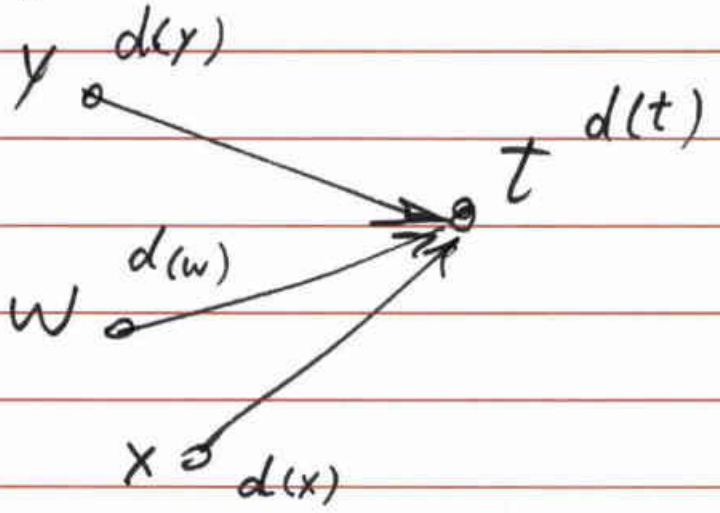
Subject to

$$0 \leq \sum_{i=1}^{m} f_i(e) \leq c_e \quad \text{for each } e \in E$$

$$\sum_{i=1}^{m} \sum_{e \text{ into } v} f_i(e) = \sum_{i=1}^{m} \sum_{e \text{ out of } v} f_i(e) \quad \text{for each } v \in V \text{ \& for } i = 1 \text{ to } m$$

Shortest Path using LP

Shortest distance from s to v is $d(v)$ for each node v.



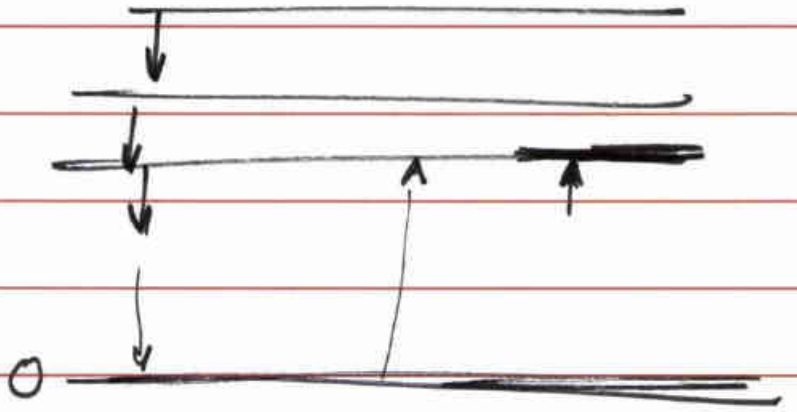
$$d(t) \leq d(y) + c_{yt}$$
$$d(t) \leq d(w) + c_{wt}$$
$$d(t) \leq d(x) + c_{xt}$$

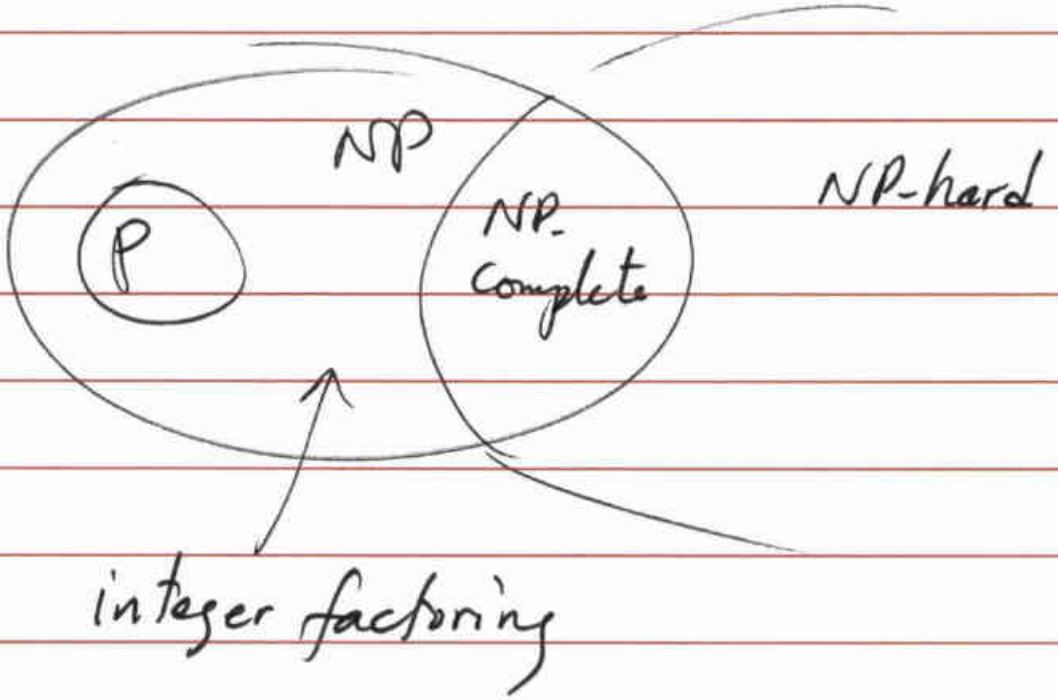
$$\begin{cases} d(v) \leq d(u) + w(u,v) & \text{for each edge } (u,v) \in E \\ d(s) = 0 \end{cases}$$

Objective function

~~Minimize~~ $d(t)$

Maximize




NP
P
NP-complete
NP-hard
integer factoring

Major difference between divide & Conquer & and dynamic programming

- Subproblems are independent/ disjoint in divide & Conquer whereas in dynamic programming They are overlapping.

Problem statement

Find the median or in general find the k^{th} smallest element of the array.

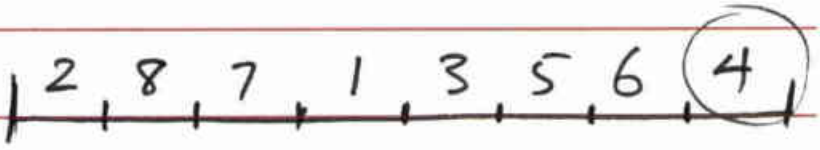
- takes $O(n \log n)$ using sorting
- can we do better!

Randomized algorithms

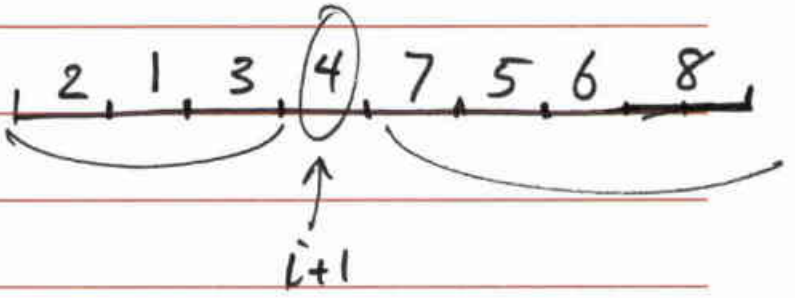
Two general types:

1- Alg. relies on random nature of input

2- Algorithm behaves randomly




```
Partion (A, p, r)
  x = A[r]
  i = p-1
  for j=p to r-1
      if A[j] ≤ x then
        i = i+1
        exchange A[i] w/ A[j]
      endif
  endfor
  exchange A[i+1] w/ A[r]
  return i+1
```



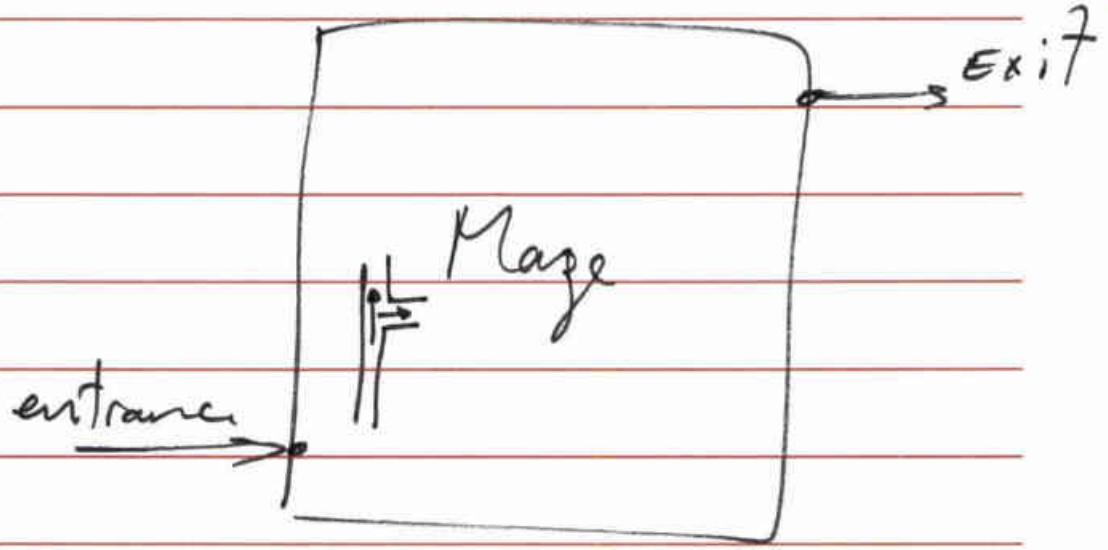
takes $O(n)$

```
Randomized-Select(A, p, r, i)
if p=r
   then return A[p]
else
   q = partion(A, p, r)
   j = q-p+1
   if i = j
      then return A[q]
   else if i < j
      then return Randomized-select(
                  A, p, q-1, i)
   else return
      Randomized select(A, q+1, r, i-j)
```

Complexity $T(n) = T\left(\frac{n}{1+\alpha}\right) + O(n)$

$n^{\log_b^q} = n^{\log_{1+\alpha}^1} = n^0$

$f(n) = O(n)$ $\Rightarrow O(n)$


Exit
Maze
entrance