$$D^{(\ell)}(i,j) = \begin{cases} \max\limits_{i\le k\le j}\left(D^{(\ell/2)}(i,k) + D^{(\ell/2)}(k,j)\right) & \text{if } \ell \text{ even} \\ \max\limits_{i\le k\le j}\left(D^{(\ell-1)}(i,k) + d(k,j)\right) & \text{if } \ell \text{ odd} \end{cases}$$

only need $O(\log L)$ values of ℓ.

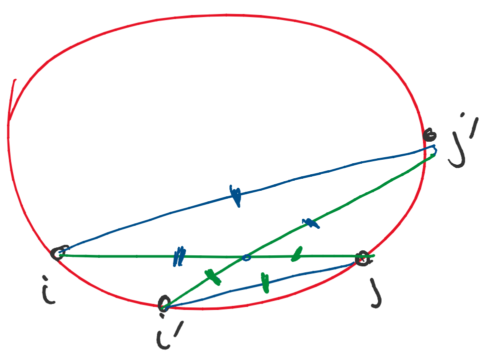
$\Rightarrow O(n^3 \log L)$ time

('repeated squaring)

Improved DP Method 2:

Def Function d satisfies convex Monge property if $\forall\, i \le i' \le j \le j'$,

$$d(i,j) + d(i',j') \geq d(i,j') + d(i',j)$$



for our problem, it follows by triangle ineq.

Lemma (F. Yao '82)

Let $D(i,j) = \max\limits_{i\le k\le j}\left(d(i,k) + d(k,j)\right)$

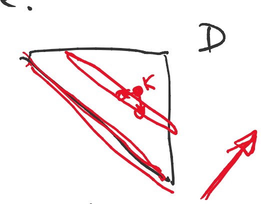
$K(i,j) =$ k that attains this max

If d is convex Monge,

then a) K is monotone increas. in each row & in each column

b) D is also convex Monge.

Cor If d is convex Monge, compute D in $O(n^2)$ time.



Pf: Fix diagonal $j = i+\Delta$.

Given $D(i, i+\Delta)$ for all i.

$K(i, i+\Delta)$

Given $D(i, i+\Delta)$ for all i.
$K(i, i+\Delta)$

Want to compute $D(i, i+\Delta+1)$ for all i
$K(i, i+\Delta+1)$

Observe that $K(i, i+\Delta) \leq K(i, i+\Delta+1) \leq K(i+1, i+\Delta+1)$
by monotonicity (Lemma (a))

$$D(i, i+\Delta+1) = \max_{K(i,i+\Delta) \leq k \leq K(i+1, i+\Delta+1)} \left(d(i,k) + d(k,j)\right)$$

in total time $O\left(\sum_i \left(K(i+1, i+\Delta+1) - K(i, i+\Delta) + 1\right)\right)$

$$\left(K(\cancel{2, 2+\Delta}) - K(1, 1+\Delta)\right) + \left(K(\cancel{3, 3+\Delta}) - K(\cancel{2, 2+\Delta})\right) + \ldots$$

telescoping sum

$= O(n)$

$\Rightarrow$ grand total over all Δ is $O(n^2)$. □

$\Rightarrow$ $\boxed{O(n^2 \log L)}$ time

Rmk - Schieber '98 $O\left(n\, c^{\sqrt{\log L \log\log n}}\right)$
$\leq O(n^{1+\varepsilon})$ for any const $\varepsilon > 0$

Optimal Binary Search Tree

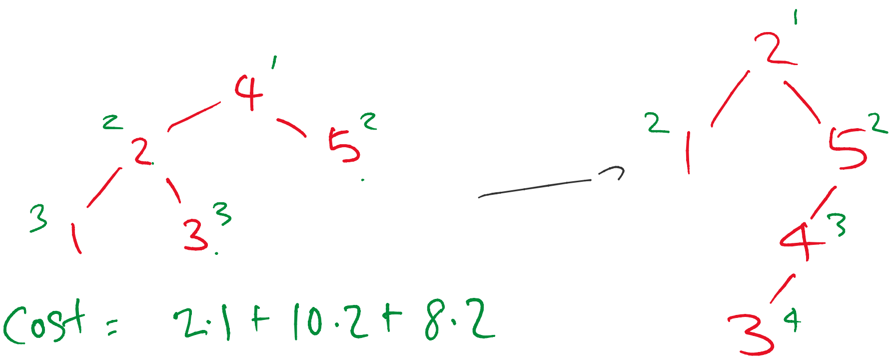
Given n values $a_1, \ldots, a_n$,

Given n values $a_1, \dots, a_n$,
& their frequencies $f_1, \dots, f_n$,

build a binary search tree T for $a_1, \dots, a_n$
that minimizes total search cost

$$\sum_{i=1}^{n} f_i \cdot \underline{\text{depth}_T(a_i)}$$

e.g. a: 1, 2, 3, 4, 5
f: 4, 10, 1, 2, 8



Cost = $2 \cdot 1 + 10 \cdot 2 + 8 \cdot 2 + 4 \cdot 3 + 1 \cdot 3 = 53$

cost = $10 \cdot 1 + 4 \cdot 2 + 8 \cdot 2 + 2 \cdot 3 + 1 \cdot 4 = 44$

18
16
10

Sort $a_1 < \dots < a_n$.

Define subproblems:

for each $1 \le i \le j \le n$,
let $D(i,j) =$ min total search cost for $a_i, a_{i+1}, \dots, a_j$

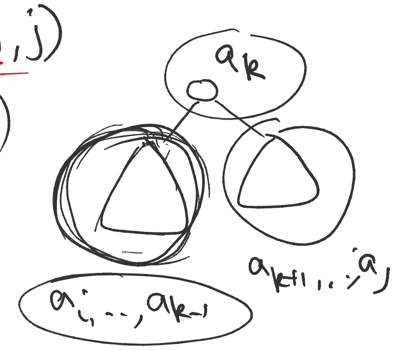
Want $D(1,n)$.

Recursive formula:

$D(i, k-1) + D(k+1, j)$

Recursive formula:

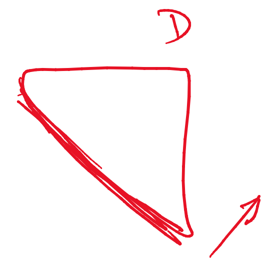
$$D(i,j) = \min_{i \le k \le j} \left(D(i, k-1) + D(k+1, j) + \underbrace{f_i + \dots + f_j}_{d(i,j)} \right)$$



Base case:

$D(i, i-1) = 0.$

Evaluation order: increasing $j-i$.



Analysis:

\# table entries $O(n^2)$

each entry in $O(n)$ time

$\Rightarrow$ total time $\boxed{O(n^3)}$

Improvement: (Knuth '71 / F. Yao '82)

Lemma Let $K(i,j) = k$ that attains min

If d is concave Monge,

then K is monotone per row / column

Same trick as in Cor $\Rightarrow$ $O(n^2)$ time

Rmk: $d(i,j) = f_i + \dots + f_j$

Monge?

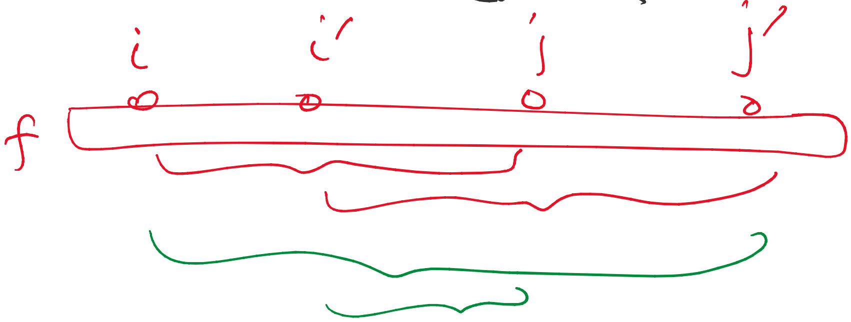
$i \le i' \le j \le j'$:

$$d(i,j) + d(i',j') = d(i,j') + d(i',j)$$

convex & concave Monge

$i \quad i' \quad j \quad j'$

convex & concave Monge



Knapsack

Given n items where item i has weight w_i ← assume integers
& value v_i

and number W (capacity),

find $I \subseteq \{1, \dots, n\}$ s.t. $\sum_{i \in I} w_i \le W$

maximizing $\sum_{i \in I} v_i$.

0/1 Version: each item used at most once (I is a set)

Unbounded version: each item may be used more than once (I is a multiset)

both (weak) NP-complete!

but if W is not too big, can use DP.

Define subproblems:

for each $i = 0, \dots, n$, $j = 0, \dots, W$,

let $C(i,j) = \max \sum_{i \in I} v_i$

s.t. $\sum_{i \in I} w_i \le j$, $I \subseteq \{1, \dots, i\}$

Want $C(n,W)$.

Recursive formula: (0/1 Version)

$$\boxed{C(i,j) = \max\{\underbrace{C(i-1,j)}_{\text{not use } i},\ \underbrace{C(i-1,j-w_i)+v_i}_{\text{use } i}\}}$$

$C(0,j) = 0$.

increas. i

$\Rightarrow$ $\boxed{O(nW)}$ time

Unbdd version: $C(i,j) = \max\{C(i-1,j),\ C(i,j-w_i)+v_i\}$

$\Rightarrow$ $O(nW)$ time

Alternate DP for unbdd:

let $C(j) = C(n,j)$

$$\boxed{C(j) = \max_{i:\, w_i \le j} \left(C(j-w_i)+v_i\right)}$$

$\Rightarrow$ $O(nW)$ time