

diagonal
(for 0)

$$\begin{aligned} & \left(i=1 \quad \overbrace{f(i+1)} - \overbrace{f(i)} \right) / \\ & = \\ & \quad f(2) - f(1) + \overbrace{f(3)} - \overbrace{f(2)} + \overbrace{f(4)} - \overbrace{f(3)} \\ & \quad \quad \quad + - \\ & = O(n) \end{aligned}$$

$\Rightarrow$ total time over all $\square$ $\boxed{O(n^2)}$. $\square$

Total time of whole algm $\boxed{O(n^2 \log L)}$.

Rmk: Schieber '98 $O(n c^{\sqrt{\log L \log \log n}})$
 $\leq O(n^{1+\epsilon})$ for any const $\epsilon > 0$.

Optimal Binary Search Tree

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

build a binary search tree for $a_1, a_2, \dots, a_n$
 that minimizes total search cost

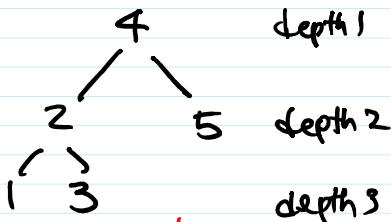
$$\sum_{i=1}^n f_i \cdot \text{depth}(a_i)$$

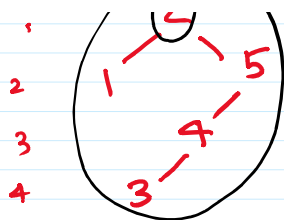
eg. $a = (1, 2, 3, 4, 5)$
 $f = (4, 10, 1, 2, 8)$

$$\begin{aligned} \text{cost} &= 4 \cdot 3 + 10 \cdot 2 + 1 \cdot 3 + 2 \cdot 1 + 8 \cdot 2 \\ &= 55 \end{aligned}$$



$$\begin{aligned} \text{cost} &= 4 \cdot 2 + 10 \cdot 1 + 1 \cdot 4 + 2 \cdot 3 + 8 \cdot 2 \\ &= 44 \end{aligned}$$





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

Sort $a_1 < a_2 < \dots < a_n$.

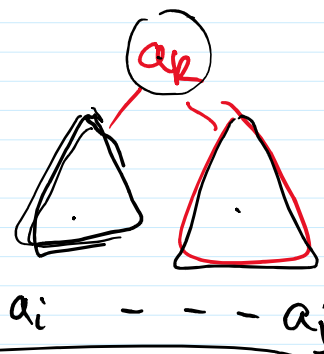
Define subproblems:

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

Want $D(1, n)$.

Recursive formula:

$$D(i, j) = D(i, k-1) + D(k+1, j) + f_i + \dots + f_{k-1} + f_{k+1} + \dots + f_j + f_k$$



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

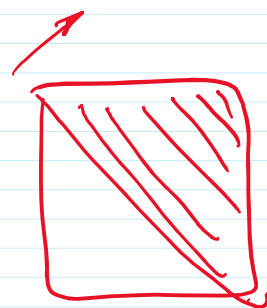
Base Case: $D(i, i-1) = 0$.

Evaluation order: increas. of $j-i$

Analysis: # table entries = $O(n^2)$

time per entry = $O(n)$

$\Rightarrow$ total time $O(n^3)$



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

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

Thm (F. Yao '82, 2nd version)

Suppose $D(i,j) = \min_{i \leq k \leq j} (D(i,k) + D(k,j) + d(i,j))$

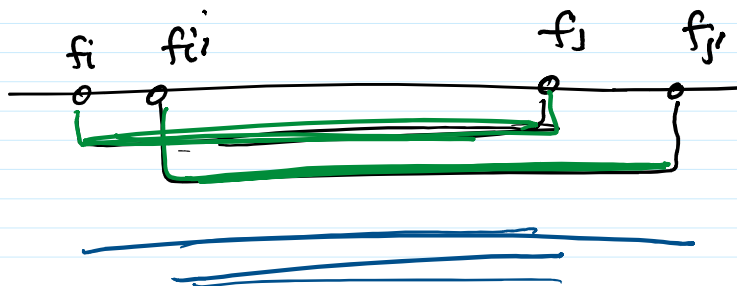
Suppose d is concave Monge.

Then can compute D in $O(n^2)$ time.

Obs $d(i,j) = f_i + \dots + f_j$.
is both convex & concave Monge.

Pf: $i < i' < j < j'$

$$\underline{d(i,j)} + \underline{d(i',j')} = \underline{d(i,j')} + \underline{d(i',j)} \quad \square$$



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

Subset Sum

Given set S of n positive integers & target number T ,
decide whether $\exists$ subset $R \subseteq S$ that sums

decide whether $\exists$ subset $R \subseteq S$ that sums to exactly T .

e.g. $\{14, 16, 20, 25, \underline{40}, 55, \underline{60}\}$

$T = \underline{100}$

$20 + 40 + 40$

Yes

Trivial algm: $O(2^n)$ time

NP-complete

$O(2^{n/2}) \approx O(\underline{1.414}^n)$

→ standard version: no repeat
multiset version: allow repeat

$\leq n/2$ | $\leq n/2$

DP Method

Let $S = \{s_1, \dots, s_n\}$.

Define subproblems: $\forall i \in \{0, \dots, n\}, t \in \{0, \dots, T\}$,

$C[i, t] = \text{yes}$ iff $\exists$ subset $R \subseteq \{s_1, \dots, s_i\}$ that sums to t

Want $C[n, T]$.

Recursive formula:

$$C[i, t] = \underbrace{C[i-1, t]}_{\text{not use } s_i} \vee \underbrace{C[i-1, t-s_i]}_{\text{use } s_i}$$

Eval order: increas. i

Analysis: # entries $O(nT)$
time per entry $O(1)$

$\Rightarrow \overline{O(nT)}$ time

← pseudopolynomial

$\Rightarrow \boxed{O(nT)}$ time $\leftarrow$ pseudopolynomial

multiset version: $C[i, t] = C[i-1, t] \vee \bigvee_k C[i-k, t-k s_i]$

$$C[i, t] = C[i-1, t] \vee C[i, t-s_i]$$

DP Method 2 (multiset version)

Define $C[t] = \text{yes}$ iff $\exists$ multiset $R \subseteq S$ that sums to t .

Formula:
$$C[t] = \bigvee_{\substack{i \in \{1, \dots, n\} \\ \text{with } s_i \leq t}} C[t-s_i]$$

entries $O(T)$

time per entry $O(n) \Rightarrow \boxed{O(nT)}$ time

DP Method 3 (multiset vers. only)

Define subproblems: $\forall t \in \{0, \dots, T\}, l,$

define $C^{(l)}(t) = \text{yes}$ iff $\exists$ multiset $R \subseteq S$ using $\leq l$ elems that sums to t .

want $C^{(l)}(T)$.

Formula:
$$C^{(l)}(t) = \bigvee_{t'=0}^t \left(\underbrace{C^{(l/2)}(t')}_{0 \leq t' \leq n} \wedge \underbrace{C^{(l/2)}(t-t')}_{0 \leq t-t' \leq n} \right)$$

$$t' = 0$$

l even

different l is $O(\log T)$

entries $O(T \log T)$

time per entry $O(T)$

$\Rightarrow O(T^2 \log T)$ worse!

$$\sum_{t'=0}^t c^{(l/2)}(t') \cdot c^{(l/2)}(t-t')$$

IS CONVOLUTION!

$\Rightarrow$ can compute $c^{(l)}$ from $c^{(l/2)}$
 $O(T \log T)$ time

$\Rightarrow$ total time $O(T \log^2 T)$

Rmk: standard vers. is tougher $O(nT)$

Koilaris & Xu '17:	$\tilde{O}(T\sqrt{n})$ time
Brüngmann '17:	$O(T \log^c T)$ rand. time
Jin & Xu '19:	$O(T \log^2 T)$ rand.
C. '26:	$O(T \log^c T)$ det. time