2025 Algorithm hw5

109062327 Hsu, Hung-Che

June 4, 2025

Problem 1: Finding All Approximate Medians in Linear Time

Given an unsorted sequence S of n distinct integers, an element $x \in S$ is an *approximate median* if

$$\left|\{ y \in S : y < x \}\right| \geq \frac{n}{4} \quad \text{and} \quad \left|\{ y \in S : y > x \}\right| \geq \frac{n}{4}.$$

Design an $O(n)$-time algorithm to find all approximate medians.

Algorithm and Pseudocode

We will use the linear-time selection subroutine to find the element of rank $\lfloor n/4 \rfloor$ (25th percentile) and the element of rank $\lceil 3n/4 \rceil$ (75th percentile), then scan S.

Algorithm 1 FindApproximateMedians(S)

```
low ← Select(S, ⌊n/4⌋) {25th percentile}
high ← Select(S, ⌈3n/4⌉) {75th percentile}
result ← []
for each x ∈ S do
   if low ≤ x ≤ high then
      append x to result
   end if
end for
return result
```

Correctness

Any element whose rank lies between $\lfloor n/4 \rfloor$ and $\lceil 3n/4 \rceil$ has at least $n/4$ elements smaller and at least $n/4$ elements larger. Scanning collects exactly those elements.

Time Complexity

Each `Select` call is $O(n)$; two selections plus an $O(n)$ scan yields $O(n)$ total.

Problem 2: Resource Allocation via Dynamic Programming

We have 4 projects and 4 resources. The profit matrix P (profit of assigning resource j to project i) is:

Project\Resource	1	2	3	4
1	3	7	10	12
2	1	2	6	9
3	2	4	8	9
4	4	2	7	10

We want to assign exactly one distinct resource to each project to maximize total profit.

DP Idea

Let $dp[i][S]$ be the maximum total profit for assigning resources to projects $1, \ldots, i$ using exactly the set S of resources. Here $S \subseteq \{1, 2, 3, 4\}$ and $|S| = i$. The recurrence is

$$dp[i][S] \;=\; \max_{r \in S} \Big\{ \, dp[i-1][\, S \setminus \{r\} \,] + P[i][r] \Big\},$$

with base $dp[0][\emptyset] = 0$. After filling $dp[1][\,\{1\}], \ldots, dp[4][\{1,2,3,4\}]$, $dp[4][\{1,2,3,4\}]$ is the optimal total profit, and we backtrack to find the assignment.

Computation of $dp[i][S]$

- **$i = 1$ ($|S| = 1$):**

$$\begin{aligned} dp[1][\{1\}] &= 3, \quad & dp[1][\{2\}] &= 7, \\ dp[1][\{3\}] &= 10, \quad & dp[1][\{4\}] &= 12. \end{aligned}$$

- **$i = 2$ ($|S| = 2$):** For each $S = \{r_1, r_2\}$, choose r for project 2 and add to $dp[1][\{\text{other}\}]$:

$$\begin{aligned}
dp[2][\{1,2\}] &= \max\big\{ \, dp[1][\{2\}] + P[2,1] = 7 + 1 = 8, \\
&\qquad\quad dp[1][\{1\}] + P[2,2] = 3 + 2 = 5 \big\} = 8, \\
dp[2][\{1,3\}] &= \max\big\{ \, dp[1][\{3\}] + P[2,1] = 10 + 1 = 11, \\
&\qquad\quad dp[1][\{1\}] + P[2,3] = 3 + 6 = 9 \big\} = 11, \\
dp[2][\{1,4\}] &= \max\big\{ \, dp[1][\{4\}] + P[2,1] = 12 + 1 = 13, \\
&\qquad\quad dp[1][\{1\}] + P[2,4] = 3 + 9 = 12 \big\} = 13, \\
dp[2][\{2,3\}] &= \max\big\{ \, dp[1][\{3\}] + P[2,2] = 10 + 2 = 12, \\
&\qquad\quad dp[1][\{2\}] + P[2,3] = 7 + 6 = 13 \big\} = 13, \\
dp[2][\{2,4\}] &= \max\big\{ \, dp[1][\{4\}] + P[2,2] = 12 + 2 = 14, \\
&\qquad\quad dp[1][\{2\}] + P[2,4] = 7 + 9 = 16 \big\} = 16, \\
dp[2][\{3,4\}] &= \max\big\{ \, dp[1][\{4\}] + P[2,3] = 12 + 6 = 18, \\
&\qquad\quad dp[1][\{3\}] + P[2,4] = 10 + 9 = 19 \big\} = 19.
\end{aligned}$$

- **$i = 3$ ($|S| = 3$):** For $S = \{r_1, r_2, r_3\}$, choose r for project 3:

$$dp[3][\{1,2,3\}] = \max\left\{\begin{aligned} &dp[2][\{2,3\}] + P[3,1] = 13 + 2 = 15,\\ &dp[2][\{1,3\}] + P[3,2] = 11 + 4 = 15,\\ &dp[2][\{1,2\}] + P[3,3] = 8 + 8 = 16 \end{aligned}\right\} = 16,$$

$$dp[3][\{1,2,4\}] = \max\left\{\begin{aligned} &dp[2][\{2,4\}] + P[3,1] = 16 + 2 = 18,\\ &dp[2][\{1,4\}] + P[3,2] = 13 + 4 = 17,\\ &dp[2][\{1,2\}] + P[3,4] = 8 + 9 = 17 \end{aligned}\right\} = 18,$$

$$dp[3][\{1,3,4\}] = \max\left\{\begin{aligned} &dp[2][\{3,4\}] + P[3,1] = 19 + 2 = 21,\\ &dp[2][\{1,4\}] + P[3,3] = 13 + 8 = 21,\\ &dp[2][\{1,3\}] + P[3,4] = 11 + 9 = 20 \end{aligned}\right\} = 21,$$

$$dp[3][\{2,3,4\}] = \max\left\{\begin{aligned} &dp[2][\{3,4\}] + P[3,2] = 19 + 4 = 23,\\ &dp[2][\{2,4\}] + P[3,3] = 16 + 8 = 24,\\ &dp[2][\{2,3\}] + P[3,4] = 13 + 9 = 22 \end{aligned}\right\} = 24.$$

- **$i = 4$ ($|S| = 4$):** Finally $S = \{1,2,3,4\}$:

$$dp[4][\{1,2,3,4\}] = \max\left\{\begin{aligned} &dp[3][\{2,3,4\}] + P[4,1] = 24 + 4 = 28,\\ &dp[3][\{1,3,4\}] + P[4,2] = 21 + 2 = 23,\\ &dp[3][\{1,2,4\}] + P[4,3] = 18 + 7 = 25,\\ &dp[3][\{1,2,3\}] + P[4,4] = 16 + 10 = 26 \end{aligned}\right\} = 28.$$

  Thus the maximum total profit is 28, attained by choosing resource 1 for project 4.

Recovering the Optimal Assignment

1. For $i = 4$, $dp[4][\{1,2,3,4\}] = 28$ comes from
$$dp[3][\{2,3,4\}] + P[4,1] = 24 + 4 = 28,$$
so project 4 uses resource 1.
2. Then for $i = 3$ with $S = \{2,3,4\}$, $dp[3][\{2,3,4\}] = 24$ comes from
$$dp[2][\{2,4\}] + P[3,3] = 16 + 8 = 24,$$
so project 3 uses resource 3.
3. For $i = 2$ with $S = \{2,4\}$, $dp[2][\{2,4\}] = 16$ comes from
$$dp[1][\{2\}] + P[2,4] = 7 + 9 = 16,$$
so project 2 uses resource 4.
4. Finally for $i = 1$ with $S = \{2\}$, $dp[1][\{2\}] = 7$, so project 1 uses resource 2.

Final Assignment and Total Profit

$$\begin{aligned}
&\text{Project } 1: \text{ resource } 2, \quad P[1,2]=7,\\
&\text{Project } 2: \text{ resource } 4, \quad P[2,4]=9,\\
&\text{Project } 3: \text{ resource } 3, \quad P[3,3]=8,\\
&\text{Project } 4: \text{ resource } 1, \quad P[4,1]=4.
\end{aligned}$$

Total profit $= 7+9+8+4 = 28$.

Problem 3: Optimal Binary Search Tree

We have $n=6$ keys $a_1,\dots,a_6$ with access probabilities

$$p_1=0.30,\ p_2=0.20,\ p_3=0.05,\ p_4=0.20,\ p_5=0.10,\ p_6=0.15,$$

and dummy keys $d_0,\dots,d_6$ of zero probability.

Define

$$e[i,j]=\min_{r=i}^{j}\big\{\, e[i,r-1]+e[r+1,j]+w[i,j]+1\big\}, \quad w[i,j]=\sum_{k=i}^{j} p_k,$$

with base $e[i,i-1]=0$ for $1\le i\le 7$.

Compute $w[i,j]$

$$\begin{aligned}
&w[1,1]=0.30, \quad w[2,2]=0.20, \quad w[3,3]=0.05, \quad w[4,4]=0.20,\\
&w[5,5]=0.10, \quad w[6,6]=0.15, \quad w[1,2]=0.50, \quad w[2,3]=0.25,\\
&w[3,4]=0.25, \quad w[4,5]=0.30, \quad w[5,6]=0.25, \quad w[1,3]=0.55,\\
&w[2,4]=0.45, \quad w[3,5]=0.35, \quad w[4,6]=0.45, \quad w[1,4]=0.75,\\
&w[2,5]=0.55, \quad w[3,6]=0.50, \quad w[1,5]=0.85, \quad w[2,6]=0.70,\\
&w[1,6]=1.00.
\end{aligned}$$

Fill $e[i,j]$ by Increasing Length

Length $\ell=0$ ($i=j+1$):

$$e[i,i-1]=0, \quad i=1,\dots,7.$$

Length $\ell=1$ ($i=j$):

$$e[i,i]=e[i,i-1]+e[i+1,i]+w[i,i]+1=0+0+p_i+1=1+p_i.$$

Hence:

$$e[1,1]=1.30,\ e[2,2]=1.20,\ e[3,3]=1.05,\ e[4,4]=1.20,\ e[5,5]=1.10,\ e[6,6]=1.15.$$

Length $\ell=2$ ($j=i+1$):

$$e[i,i+1]=\min_{r=i}^{i+1}\{\, e[i,r-1]+e[r+1,i+1]+w[i,i+1]+1\}.$$

$$
\begin{aligned}
e[1,2]:\ w[1,2] = 0.50,\ & r = 1 : 0 + 1.20 + 0.50 + 1 = 2.70,\\
& r = 2 : 1.30 + 0 + 0.50 + 1 = 2.80,\\
& \implies e[1,2] = 2.70.\\
e[2,3]:\ w[2,3] = 0.25,\ & r = 2 : 0 + 1.05 + 0.25 + 1 = 2.30,\\
& r = 3 : 1.20 + 0 + 0.25 + 1 = 2.45,\\
& \implies e[2,3] = 2.30.\\
e[3,4]:\ w[3,4] = 0.25,\ & r = 3 : 0 + 1.20 + 0.25 + 1 = 2.45,\\
& r = 4 : 1.05 + 0 + 0.25 + 1 = 2.30,\\
& \implies e[3,4] = 2.30.\\
e[4,5]:\ w[4,5] = 0.30,\ & r = 4 : 0 + 1.10 + 0.30 + 1 = 2.40,\\
& r = 5 : 1.20 + 0 + 0.30 + 1 = 2.50,\\
& \implies e[4,5] = 2.40.\\
e[5,6]:\ w[5,6] = 0.25,\ & r = 5 : 0 + 1.15 + 0.25 + 1 = 2.40,\\
& r = 6 : 1.10 + 0 + 0.25 + 1 = 2.35,\\
& \implies e[5,6] = 2.35.
\end{aligned}
$$

Length $\ell = 3$ $(j = i + 2)$:

$$
e[i, i+2] = \min_{r=i}^{i+2}\{\, e[i, r-1] + e[r+1, i+2] + w[i, i+2] + 1\}.
$$

$$
\begin{aligned}
e[1,3]:\ w[1,3] = 0.55,\ & r = 1 : 0 + 2.30 + 0.55 + 1 = 3.85,\\
& r = 2 : 1.30 + 1.05 + 0.55 + 1 = 3.90,\\
& r = 3 : 2.70 + 0 + 0.55 + 1 = 4.25,\\
& \implies e[1,3] = 3.85.\\
e[2,4]:\ w[2,4] = 0.45,\ & r = 2 : 0 + 2.30 + 0.45 + 1 = 3.75,\\
& r = 3 : 1.20 + 1.20 + 0.45 + 1 = 3.85,\\
& r = 4 : 2.30 + 0 + 0.45 + 1 = 3.75,\\
& \implies e[2,4] = 3.75.\\
e[3,5]:\ w[3,5] = 0.35,\ & r = 3 : 0 + 2.40 + 0.35 + 1 = 3.75,\\
& r = 4 : 1.05 + 1.10 + 0.35 + 1 = 3.50,\\
& r = 5 : 2.30 + 0 + 0.35 + 1 = 3.65,\\
& \implies e[3,5] = 3.50.\\
e[4,6]:\ w[4,6] = 0.45,\ & r = 4 : 0 + 2.35 + 0.45 + 1 = 3.80,\\
& r = 5 : 1.20 + 1.15 + 0.45 + 1 = 3.80,\\
& r = 6 : 2.40 + 0 + 0.45 + 1 = 3.85,\\
& \implies e[4,6] = 3.80.
\end{aligned}
$$

Length $\ell = 4$ $(j = i + 3)$:

$$
e[i, i+3] = \min_{r=i}^{i+3}\{\, e[i, r-1] + e[r+1, i+3] + w[i, i+3] + 1\}.
$$

$$
\begin{aligned}
e[1,4]:\ w[1,4]=0.75,\ & r=1: 0+3.75+0.75+1=5.50,\\
& r=2: 1.30+2.30+0.75+1=5.35,\\
& r=3: 2.70+1.20+0.75+1=5.65,\\
& r=4: 3.85+0+0.75+1=5.60,\\
& \implies e[1,4]=5.35.\\
e[2,5]:\ w[2,5]=0.55,\ & r=2: 0+3.50+0.55+1=5.05,\\
& r=3: 1.20+2.40+0.55+1=5.15,\\
& r=4: 2.30+1.10+0.55+1=4.95,\\
& r=5: 3.75+0+0.55+1=5.30,\\
& \implies e[2,5]=4.95.\\
e[3,6]:\ w[3,6]=0.50,\ & r=3: 0+3.80+0.50+1=5.30,\\
& r=4: 1.05+2.35+0.50+1=4.90,\\
& r=5: 2.30+1.15+0.50+1=4.95,\\
& r=6: 3.50+0+0.50+1=5.00,\\
& \implies e[3,6]=4.90.
\end{aligned}
$$

Length $\ell = 5$ ($j = i + 4$):

$$
e[i,i+4]=\min_{r=i}^{i+4}\{\, e[i,r-1]+e[r+1,i+4]+w[i,i+4]+1\}.
$$

$$
\begin{aligned}
e[1,5]:\ w[1,5]=0.85,\quad & r=1: 0+4.95+0.85+1=6.80,\\
& r=2: 1.30+3.50+0.85+1=6.65,\\
& r=3: 2.70+2.40+0.85+1=6.95,\\
& r=4: 3.85+1.10+0.85+1=6.80,\\
& r=5: 5.35+0+0.85+1=7.20,\\
& \implies e[1,5]=6.65.\\
e[2,6]:\ w[2,6]=0.70,\quad & r=2: 0+4.90+0.70+1=6.60,\\
& r=3: 1.20+3.80+0.70+1=6.70,\\
& r=4: 2.30+2.35+0.70+1=6.35,\\
& r=5: 3.75+1.15+0.70+1=6.60,\\
& r=6: 4.95+0+0.70+1=6.65,\\
& \implies e[2,6]=6.35.
\end{aligned}
$$

Length $\ell = 6$ ($i = 1, j = 6$):

$$
e[1,6]=\min_{r=1}^{6}\{\, e[1,r-1]+e[r+1,6]+w[1,6]+1\},\quad w[1,6]=1.00.
$$

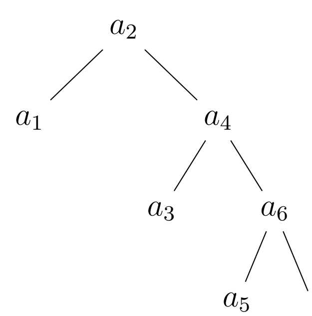
$$
\begin{aligned}
r = 1 &: 0 + 6.35 + 1.00 + 1 = 8.35, \\
r = 2 &: 1.30 + 4.90 + 1.00 + 1 = 8.20, \\
r = 3 &: 2.70 + 3.80 + 1.00 + 1 = 8.50, \\
r = 4 &: 3.85 + 2.35 + 1.00 + 1 = 8.20, \\
r = 5 &: 5.35 + 1.15 + 1.00 + 1 = 8.50, \\
r = 6 &: 6.65 + 0 + 1.00 + 1 = 8.65, \\
&\Longrightarrow e[1, 6] = 8.20.
\end{aligned}
$$

The minimum 8.20 is attained by $r = 2$ or $r = 4$. Choose $r = 2$ at the root.

Recovering the BST Structure

- Root of full tree: a_2 (since $r = 2$ gave 8.20).
- Left subtree of a_2 (keys a_1): root $r = 1$, cost $e[1, 1] = 1.30$.
- Right subtree of a_2 (keys $a_3, \ldots, a_6$): cost $e[3, 6] = 4.90$, attained by $r = 4$.
  - Left subtree of a_4 (key a_3): root $r = 3$, cost $e[3, 3] = 1.05$.
  - Right subtree of a_4 (keys a_5, a_6): cost $e[5, 6] = 2.35$, attained by $r = 6$.
    * Left subtree of a_6 (key a_5): $r = 5$, cost $e[5, 5] = 1.10$.
    * Right subtree of a_6: empty, cost 0.

Thus one optimal BST is:



with expected cost $e[1, 6] = 8.20$.

Problem 4: Weighted Interval Scheduling

Given n jobs (s_i, f_i, w_i), find a maximum-weight subset of non-overlapping jobs.

Algorithm

1. Sort jobs by ascending finish time f_i.
2. For each i, compute

$$p(i) = \max\{\, j < i : f_j \le s_i \}, \quad \text{or 0 if none.}$$

3. Define $M[0] = 0$. For $i = 1, \ldots, n$, set

$$M[i] = \max\{\, M[i-1],\; M[p(i)] + w_i \}.$$

4. The answer is $M[n]$. Backtracking yields the chosen jobs.

Time Complexity

Sorting: $O(n \log n)$. Computing each $p(i)$ by binary search: $O(\log n)$ each, total $O(n \log n)$. Filling M in $O(n)$. Overall $O(n \log n)$.