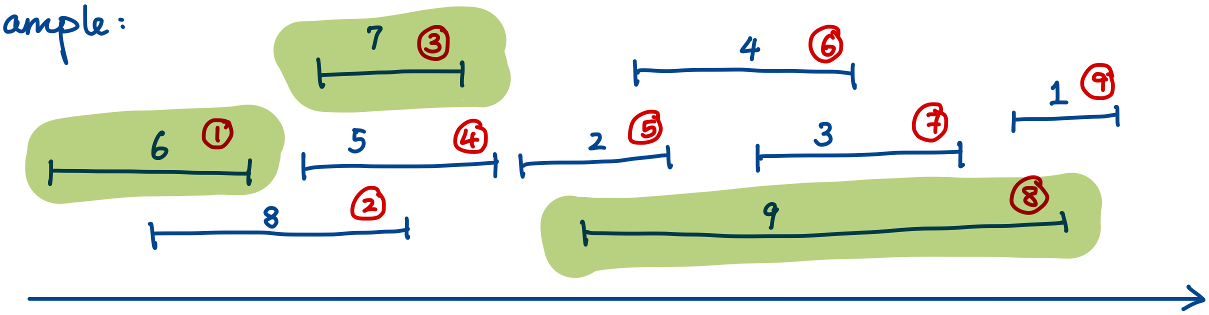
Dynamic Programming [chapter 6]

Weighted Interval Scheduling (WIS)

Example:



$P(1)=0$, $P(2)=0$, $P(3)=1$, $P(4)=1$, $P(5)=4$, $P(6)=4$,
$P(7)=5$, $P(8)=4$, $P(9)=7$.

OPT(j) - best you can do in intervals [1, 2, --- j]

$\mathrm{OPT}(0) = 0$

$\mathrm{OPT}(1) = 6$

$\mathrm{OPT}(2) = \max(8 + \mathrm{OPT}(\overset{0}{\overset{\downarrow}{P(2)}}), \mathrm{OPT}(1)) = \max(8, 6) = 8$

$\mathrm{OPT}(3) = \max(7+6, 8) = 13$

$\mathrm{OPT}(4) = \max(5+6, 13) = 13$

$\mathrm{OPT}(5) = \max(2 + \mathrm{OPT}(P(5)), \mathrm{OPT}(4)) = \max(2+13, 13) = 15$

$\mathrm{OPT}(6) = \max(4+13, 15) = 17$

$\mathrm{OPT}(7) = \max(3+15, 17) = 18$

$\mathrm{OPT}(8) = \max(9+13, 18) = 22$

$\mathrm{OPT}(9) = \max(1+18, 22) = 22$

max_weight = OPT(9) ← 22

solution set =

↓ $\{1, 3, 8\}$

To get all the interval:

Check $\mathrm{OPT}(9) > \mathrm{OPT}(8)$: No → 9 is not in the solution.

$\mathrm{OPT}(8) > \mathrm{OPT}(7)$: Yes → 8 is in the solution.

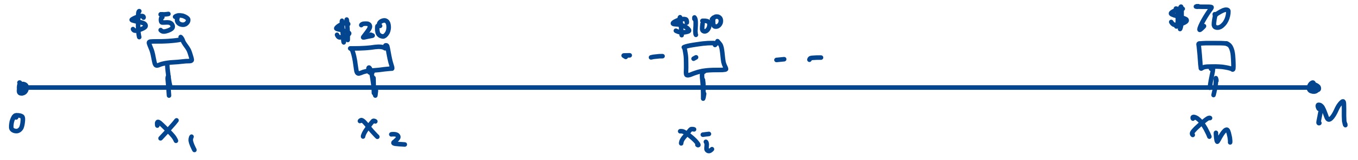
$\mathrm{OPT}(P(8)) > \mathrm{OPT}(P(8)-1)$:
↳ $\mathrm{OPT}(4) > \mathrm{OPT}(3)$ No → 4 is not in the solution

$\mathrm{OPT}(3) > \mathrm{OPT}(2)$: Yes → 3 is in the solution

$\mathrm{OPT}(P(3)) > \mathrm{OPT}(P(3)-1)$:
↳ $\mathrm{OPT}(1) > \mathrm{OPT}(0)$: Yes → 1 is in the solution.

Billboards Problem : A highway with M miles.

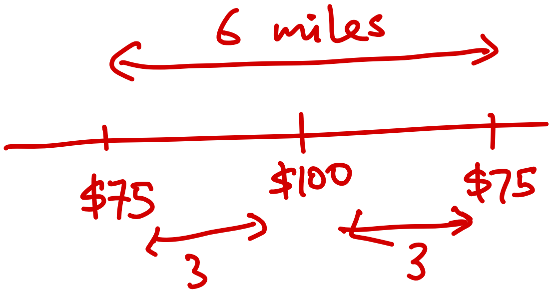
n locations $x_1, x_2, \ldots, x_n$ on $[0, M]$



If a BB is placed at x_i, we will get $\$ r_i$.

However, the road department has a rule that no 2 BBs can be placed within 5 miles of each.

How to place the billboards so as to maximize the revenue?



Dynamic Programming :

Subproblems: $1 \le i \le n$

Problem(i): Given $\{x_1, x_2, \ldots x_i\}$ and $\{r_1, r_2, \ldots r_i\}$ place BBs so as to max. revenue.

Answer to Problem(i) is $OPT(i)$ (revenue).

Recurrence Relation : $OPT(i) = f(OPT(i-1), OPT(i-2), \ldots OPT(1))$.

$$OPT(i) \begin{cases} \text{Place BB at } x_i & = r_i + OPT(\underset{\uparrow}{P(i)}) \\ \text{Don't place BB at } x_i & = OPT(i-1) \end{cases}$$

Preprocessing for each $\bar{i}$, find $p(i) =$ the max $j < \bar{i}$, s.t. x_j is > 5 miles aways from x_i.

max revenue $= OPT(n)$.

Maximum Subarray

A = [10, -1, -2, -3, 10]

Input : Array A = [-2, 4, 6, -5, -4, 2, -7, $\overset{18}{\cancel{8}}$]

1 2 3 4 5 6 7 8

S(3, 5) = -3 , S(2, 3) = 10 which is largest.

Output : Possitions i and j and S(i, j) which is largest possible.

① Define Subproblem. j = 1 to n.

OPT(j) - the sum of maximum subarray in the array A[1, 2, ... j].

Output : OPT(n). ← max sum

OPT(1)= -2 , OPT(2)= 4, OPT(3)= 10, OPT(4)=10

OPT(5)=10, OPT(6)= 10, OPT(7)=10 , OPT(8)=10

OPT(8)=18

We can't really find a recurrence relation.

Another definition of OPT(j):

OPT(j) - the sum of maximum subarray ending at possition j.

A = [-2, 4, 6, -5, -4, 2, -7, 8]

1 2 3 4 5 6 7 8

OPT(1)=-2, OPT(2)=4 OPT(3)=10, OP(4)=5

OPT(5)=1, OPT(6)=3, OPT(7)=-4 , OP(8)=8

Answer = max (OPT(1), OPT(2), ... OPT(n).

$$OPT(j) = \max(A[j] + OPT(j-1), A[j]).$$

j = the index where $A[j]$ = Answer.

i = the inden $\le j$ where $OPT(j) = A[i]$.