CMPS102 HW3

Jinxuan Jiang

Question 1 “On The Run”:

Solution:

Let OPT(n) denotes the fastest route from Santa Cruz to Achliopolis.

Then, we have two possibilities:

Case 1:

Starting from Highway A, let **FA(j)** denotes the fastest possible time starting from Highway A.

Case 2:

Starting from Highway B, let **FB(j)** denotes the fastest possible time starting from Highway B.

So, the **OPT(n) = min{ FA(n) , FB(n) }**

Let A(i) be the fastest way to the last junction A(i-1) plus time to travel between A(i-1) to A(i) or the fastest way to last junction B(i-1), plus the time to travel between B(i-1) to A(i) plus switch time k.

Same for B(i).

Then, We can calculate **FA(n)** and **FB(n)** by **recursion**:

Case 1:

If i =1,

FA(i) = A(1)

Else if i > 1

FA(i) = min{ FA(i-1) + A(n), FB(i-1)+B(n)+k}

Case 2:

If i = 1

FB(i) = B(1)

Else if i > 1

FB(i) = min{ FB(i-1) + B(n), FA(i-1)+A(n)+k}

OPT(n) = min{ FA(n) , FB(n) }

Run Time:

The recursive call takes O(n^2) time.

With memorization, the function will take O(n) time and O(n) space.

Proof of correctness: (Induction)

Let OPT(n) be the fastest route from Santa Cruz to Achliopolis.

Base case:

If i = 0, there is no junction between Highway A and B.

If i = 1, FA(1)=A(1) or FB(1)=B(1).

Obviously, the base case are true.

Inductive step:

Let n >1 and for all the n > n0,

assume OPT(n) is true, then prove OPT(n+1) is true.

Highway A

SC(0) 1 2 3 …………………………………………n-1, n, Achliopolis(n+1)

Highway B

OPT(n) = OPT(n-1) + min(distance from n-1 to n)

OPT(n-1) = OPT(n-2) +min(distance from n-2 to n-1)

…………………………………………………………...

OPT(2) = OPT(1) + min(distance from 0 to 1)

OPT(1) is the base case,

From the above infer, we know that the value of each function is dependent.

OPT(1) is optimal(base case), then plus the minimum distance from junction 0 to 1, so the OPT(2) is optimal. Then we can conclude that OPT(n) is optimal. So, OPT(n+1) = OPT(n) + minimum distance from the last junction to Achliopolis, which means OPT(n+1) is optimal.

Therefore, **OPT(n) is the fastest route from Santa Cruz to Achliopolis.**

Question 2 “ Engine Trouble” :

Solution:

Let OPT(L) denotes the cheapest way to make L repairs,

Let C(i) denotes the cost of Maggy’s repair, where C(i) = t(i) * r

The cheapest way to make L has two cases:

Case1:

Mikey repairs the last 5

Case 2:

Maggy repairs the Lth.

Then, we have the **OPT(L) = min{OPT(L-4)+B, OPT(L-1)+C(L)}**

Then we can calculate OPT(L) by recursion.

If L = 0,

OPT(L) = 0;

Else if L = 1,

OPT(L) = min{b,C(1)}

Else if L = 2,

OPT(L) = min{b,C(1)+C(2)}

Else if L = 3,

OPT(L) = min{b,C(1)+C(2)+C(3)}

Else if L = 4,

OPT(L) = min{b, C(1)+C(2)+C(3)+C(4)}

Else (L ≥ 5)

OPT(L) = min{OPT(L-4)+b, OPT(L-1)+C(L)}

Runtime：

The run time of the recursive function OPT is O(n^2).

With memorization, the function will take O(n) time and O(n) space.

Proof of correctness: (**by induction**)

Let OPT(L) be the cheapest way to make L repairs.

Base case:

If L= 0, then OPT(0) = 0.

If L =1, then OPT(1) = min{b,C(1)}, we just need to compare the cost of b and C(1), and return the smallest value.

If L =2, then OPT(1) = min{b,C(1)+C(2)}, we just need to compare the cost of b and sum of C(1), C(2) and return the smallest value.

If L =3, then OPT(1) = min{b,C(1)+C(2)+C(3)}, we just need to compare the cost of b and C(1), C(2), C(3) and return the smallest value.

If L =4, then OPT(1) = min{b,C(1)+C(2)+C(3)+C(4)}, we just need to compare the cost of b and the sum of C(1), C(2), C(3), C(4), and return the smallest value.

Obviously, all the five base cases will always return the smallest value.

Inductive step:

Assume that for all the L ≥ 5, there always exist a L for

OPT(L) = min{OPT(L-4)+b, OPT(L-1)+C(L)} is always the cheapest way to make L repairs.

As we know, all the value of OPT(L) are dependent. And all the future decision are based on the previous step.

For OPT(L-4), it would lead to

OPT(L-4) = min{OPT((L-4)-4) +b, OPT((L-4)-1)+C(L-4)}

…………………………………………………………..

Finally, it would lead to the 5 base cases above, which would always return the optimal value for the function. Because we solve the problem recursively, it will always return the optimal value. As we know, all the value of OPT(L) are dependent. And all the future decision are based on the previous step, if the previous step is optimal, the overall solution is optimal too.

Therefore, **OPT(L) be the cheapest way to make L repairs.**

Question 3 “Achliopolis Vegan Hot Dog Eating Champion”:

Solution:

Let OPT(i) denotes the maximum total earning on ith day

Then, the total earning OPT(i) = OPT(i-3)+C(i)

because you need to fast on (i-2),(i-1),(i+1),(i+2).

If you don’t enter, then the OPT(i) = OPT(i-1).

So, **OPT(i) = max{OPT(i-3)+C(i), OPT(i-1)}**

Then, we can calculate the maximum earning by recursion:

If i = 1

    OPT(i) = max{C(1)}

Else if i = 2

    OPT(i) = max{C(2), OPT(1)}

Else if i = 3

    OPT(i) = max{C(3),OPT(2),OPT(1)}

Else (i ≥ 4)

    OPT(i) = max{OPT(i-3)+C(i), OPT(i-1)}

Runtime:

The run time of the recursive function OPT is O(n^2).

With memorization, the function will take O(n) time and O(n) space.

Proof of correctness: (by induction)

Let OPT(i) be the maximum total earning on ith day.

Base case:

If i = 1, the maximum earning is c1

If i = 2, you need to compare the earning between c1 and c2, and return the maximum value.

If i = 3, you need to compare the earning between c1, c2 and c3, and return the maximum value.

Obviously, the three base cases will always return the maximum value.

Inductive Step:

Assume that there always exists a i ≥ 4 for

OPT(i) = max{OPT(i-3)+C(i), OPT(i-1)}

For OPT(i-3), it will lead to:

OPT(i-3) = max{OPT((i-3)-3)+C(i-3),OPT(i-3)-1)}

…………………………………………………………………...

Finally, it would lead to the 3 base cases above, which would always return the optimal value for the function. Because we solve the problem recursively, it will always return the optimal value. As we know, all the value of OPT(i) are dependent. And all the future decision are based on the previous step, if the previous step is optimal, the overall solution is optimal too.

Therefore, **OPT(i) be the maximum total earning on ith day.**