Qiyao Zhou

Z5379852

Question 2

From this problem, we can use dynamic programming in the question. During dynamic planning, we save the result every time we calculate the distance between two camps that has not been calculated before, or we can call it directly if it has been calculated before.

Subproblems: We can solve this problem by considering the subproblem P(i) = $[A_m, B_n]$: For every 0<=i<=2n+1, P(i) means the location of Alice and Bob's respective arrivals on day i.

Recurrence: For 0<=i<=2n+1, P (i) = $[A_x, B_y]$,then we can compare the distance between d_1=$[A_{x+1}, B_y]$, d_2= $[A_x, B_{y+1}]$ and d_3= $[A_{x+1}, B_{y+1}]$, P (i+1) = argmin(d_1, d_2, d_3).

Base case: P (0) = $[A_1, B_1]$.

Order of computation: subproblem P(i) depends only on earlier subproblems (P(j), where j < i), so we can solve the subproblems in increasing order of i.

Final answer: Finally we get the best path choice P(0) to P(last). Then we can create D=d(P(0)) and compare D with d(P(i)) for i from 1 to last. If D< d(P(i)) then D= d(P(i)), else continue.

In the provided example, after this algorithm D is 3 which corresponds to reality.

Time complexity: As for each distance $[A_x,\ B_y]$ we only compute once using dynamic programming, 1<=x<=n, 1<=y<=n. So, we need to compute n^2 times. Every time we need o (1). And the process of comparison cost o (n) so the overall time complexity is o (n^2).