**D13MAXR**

* Sub 1 : duyệt hoán vị
* Sub 2 : DP bitmask

- Sub 3 :

**- Nhận xét:** Nếu như k là vị trí cuối cùng dc xóa trong đoạn (l, r) thì nó đã xóa hết những cái từ (l, k) và (k, r)

=> Đệ quy có nhớ.

- Kết quả cập nhật cho f(l, r) là max ( (l , k) +

(k, r) + a[l] \*a[r]) (l < k< r).

**D13ROBOT**

* Sub 1: LU thì mọi robot luôn đi dc về đích.
* Sub 2: DFS tìm đường đi đến (0 , 0).
* Sub 3: n lần LU luôn đi dc về đích.
* Sub 4: BFS(x, y, v, v).
* Sub 5: Đưa robot 1 về (0 , 0) lúc đó thì robot 2

(c , d) -> (c’ , d’) làm tương tự với robot2 các bước cứ lặp lại đến khi cả 2 con ở (0, 0). (Làm được cách này bởi vì khi lặp lại ta luôn đảm bảo vị trí của mỗi con robot đều gần hơn so với trước đó nên luôn về đế đich (0, 0) ).

**CAMELOT**

* Sub 1: Dfs(x , y , u , v)
* Sub 2 + 3:
* Bfs tìm đường đi ngắn nhất của vua và mã tới mọi điểm trên bàn cờ
* Với mỗi (u , v) ô ta xét coi chênh lệch đường đi của 2 con có chia hết cho 2 không , nếu có thì
* ans = min ( max (d1[u][v] , d2[u][v] ) , ans)

\*có thể mở rộng cho bài toán đồ thị vô hướng

Dfs(i, j, z)

z là trạng thái của vua và mã nếu = 1 thì là đi 1 con rồi , nếu = 2 thì là đã đi cả 2 con