

Aluno: André Santos Rocha

RA: 235887

Problema 1: C - Kuroki and the Gifts

O problema nos fornece dois arrays de n inteiros, um desses arrays representa brilhos de colares e o outro representa brilho de pulseiras. Queremos distribuir um colar e uma pulseira para n garotas, sendo que as somas dos brilhos do colar e da pulseira de cada garota devem ser distintas dois a dois.

1.1 Ideia de solução

Para este problema, ao receber os arrays podemos ordenar ambos vetores. Daí, como todos os valores de brilho para cada acessório são distintos, sabemos que as somas serão sempre distintas, pois, cada soma s é composta de dois números P_s e C_s , tais que $P_s > P_{s-1}$ e $C_s > C_{s-1}$, sendo assim s sempre será maior do que $s-1$ e, portanto, distinta. Por fim, podemos apenas printar os vetores ordenados e o problema está resolvido.

1.2 Detalhes de implementação

Como a ordenação demanda $O(n \log n)$, essa será a complexidade do nosso problema.

Problema 2: H - Longest Palindrome

O problema nos fornece n strings. Para cada string, devemos printar o comprimento do maior palíndromo que pode ser obtido removendo 0 ou mais caracteres.

2.1 Ideia de solução

Neste problema, podemos utilizar programação dinâmica. Faremos nossa matriz de dp nos baseando em dois ponteiros l e r , sendo que l aponta para o começo da string e r para o fim. O valor de $dp[l][r]$ será o comprimento do maior palíndromo que se inicia em l e termina em r . Ao iniciar nosso algoritmo para preencher a matriz, exploramos a seguinte relação:

- Se $l > r$, já analisamos todas as possibilidades de palíndromo, então retornamos 0;
- se $l == r$, então o palíndromo é composto pela própria letra, então retornamos 1;
- se o caractere na posição l for igual ao na posição r , retornamos 2 (os dois caracteres l e r) mais $dp[l+1][r-1]$.
- se o caractere na posição l for diferente ao na posição r , retornamos o máximo entre $dp[l][r-1]$ e $dp[l+1][r]$.

Por fim, basta retornar $dp[l][r]$ e teremos o resultado que buscamos.