

Aluno: André Santos Rocha

RA: 235887

Problema 1: C - Double Strings

Dadas n strings de tamanho no máximo 8, precisamos determinar se, para cada string i , existem duas outras j e k tal que $S_i = S_j + S_k$, isto é, a concatenação de S_j e S_k .

1.1 Ideia de solução

Para este problema, podemos utilizar uma abordagem de busca completa. Criamos um set para guardar todas as strings recebidas. Depois, para cada string recebida, utilizamos um ponteiro i para iterar pelos seus caracteres, dividindo a string em dois pedaços: $string[0:i]$ e $string[i:len(string)]$. Para cada um desses pares de pedaços, checamos se o par está no conjunto. Com isso, caso encontremos um par no conjunto, quer dizer que é possível concatenar duas strings para formar a terceira string. Caso não encontremos nenhum par no conjunto, não será possível.

1.2 Detalhes de implementação

Como as strings possuem tamanho no máximo igual a 8, não é problema realizar a busca completa. Realizamos n iterações pelas strings e no máximo 8 iterações para cada string. Com isso, teremos $O(8 * n) = O(n)$.

Problema 2: E - $P \neq NP$

Neste problema, devemos contar todos os pares de inteiros (n, p) tais que $0 \leq p \leq P$, $p \neq n \cdot p$, $p! = n \cdot p$.

2.1 Ideia de solução

Neste problema, perceba que sempre dada uma entrada P , teremos $P + 1$ números indo de 0 a P . Para cada um desses números p , podemos escolher um número n que seja igual a $(p-1)!$. Exceto quando $p=0$, $p=1$ e $p=2$, pois nesses casos, $p! = n \cdot p$ implica $p = n \cdot p$. Então sabemos que será possível realizar $P + 1 - 3$ pares, o que nos dá $P - 2$. Então se P for 0, 1, ou 2, então devolvemos 0. Se P for maior que 2, então devolvemos $P - 2$.

2.2 Detalhes de implementação

Neste problema, a complexidade será $O(1)$.