

Parejas de Baile

13. Tenemos dos conjuntos de personas y para cada persona sabemos su habilidad de baile. Queremos armar la máxima cantidad de parejas de baile, sabiendo que para cada pareja debemos elegir exactamente una persona de cada conjunto de modo que la diferencia de habilidad sea menor o igual a 1 (en módulo). Además, cada persona puede pertenecer a lo sumo a una pareja de baile. Por ejemplo, si tenemos un multiconjunto con habilidades $\{1, 2, 4, 6\}$ y otro con $\{1, 5, 5, 7, 9\}$, la máxima cantidad de parejas es 3. Si los multiconjuntos de habilidades son $\{1, 1, 1, 1, 1\}$ y $\{1, 2, 3\}$, la máxima cantidad es 2.

- Considerando que ambos multiconjuntos de habilidades están ordenados en forma creciente, observar que la solución se puede obtener recorriendo los multiconjuntos en orden para realizar los emparejamientos.
- Diseñar un algoritmo goloso basado en [a\)](#) que recorra una única vez cada multiconjunto. Explicitar la complejidad temporal y espacial auxiliar.
- Demostrar que el algoritmo dado en [b\)](#) es correcto.

```
vector<int> personas1;  
vector<int> personas2;  
sort(personas1);  
sort(personas2);  
int PB(personas1, personas2)  
{  
    int count = 0;  
    int i = 0, j = 0;  
    while (i < personas1.size() & j < personas2.size())  
    {  
        if (abs(personas1[i] - personas2[j]) <= 1) then  
            count++;  
            i++; j++;  
        else if (personas1[i] > personas2[j]) then  
            j++;  
        else  
            i++;  
    }  
    return count;  
}
```

Complejidad espacial $O(\max(|personas1|, |personas2|))$
temporal $O(\max(|personas1| \log |personas1|, |personas2| \log |personas2|))$
La temporal de PB es $O(\max(|personas1|, |personas2|))$

¿Cómo se arma una solución? Comparando 2 valores y viendo si su resta en módulo ≤ 1 . En ese caso sumo 1. Caso contrario comparo otros valores.

Estrategia greedy: recorro de menor a mayor una lista de personas y fijame si la resta en módulo de las habilidades son ≤ 1 . Si no va sumando hasta encontrar

QVQ esta estrategia es correcta, es decir, nos lleva a una solución correcta.

Sea v^* una solución de nuestro problema hallada con la estrategia greedy QVQ
 v^* es correcta.

Como tenemos que sumar 1 / o sea que la diferencia de habilidades sea $\leq 1 \rightarrow$
tengo que sumar 1 \Leftrightarrow habilidad persona $nultroij A =$ habilidad persona $nultroij B \checkmark$
habilidad persona $nultroij A+1 =$ habilidad persona $nultroij B \checkmark$
habilidad persona $nultroij A =$ habilidad persona $nultroij B+1$

Como en un algoritmo A y B están representados como secuencias ordenadas, me basta con ir comparando posición a posición solamente. Si la diferencia en módulo de la pos $A[i]$ con $B[i]$ no es ≤ 1 , me fijo cuál es el que es más grande y entonces dicho fijo esa posición y voy avanzando con la otra. Sup. que $B[i] > A[i]$, puedo ir aumentando i ya que todo $k \in \mathbb{N} / k < i$ $A[k] < A[i]$ y por lo tanto $|B[i] - A[k]| > 1 \rightarrow$ no estoy perdiendo respuestas posibles.