

Esercizi sulla Tecnica Greedy.

Ugo Vaccaro

1. *Esercizio:* Si applichi l'algoritmo `Greedy_Activity_Selector` sul seguente insieme di attività: $A = \{A_1, A_2, \dots, A_{11}\}$, con $A_1 = [9, 12]$, $A_2 = [3, 14]$, $A_3 = [6, 10]$, $A_4 = [9, 13]$, $A_5 = [4, 9]$, $A_6 = [13, 15]$, $A_7 = [7, 11]$, $A_8 = [2, 5]$, $A_9 = [6, 8]$, $A_{10} = [4, 6]$, $A_{11} = [1, 7]$, illustrandone il comportamento ad ogni passo.

◇

2. *Esercizio:* Si intende riparare un tubo di gomma che ha n fori su di esso. Si rappresenti l' i -esimo ciascun foro con il numero x_i , dove x_i è la distanza, in centimetri, del foro i -esimo dalla estremità sinistra del tubo. La riparazione avviene attaccando pezzi di nastro adesivo al tubo, dove ciascun pezzo di nastro adesivo è lungo k centimetri, k parametro dato. Il problema è quello di riparare il tubo (ovvero coprire tutti i buchi) utilizzando il *minor numero* di pezzi di nastro adesivo.

Si consideri il seguente algoritmo greedy per il problema appena enunciato: si ordinino i numeri x_i in modo tale che $x_1 \leq x_2 \leq \dots \leq x_n$, indi si attacchi un pezzo di nastro adesivo al tubo, partendo dal buco x_1 fino a $x_1 + k$. Si iteri sui buchi rimasti scoperti, fin a quando tutti i buchi sono stati coperti.

Si provi che tale algoritmo copre tutti i buchi utilizzando minor numero di pezzi di nastro adesivo.

◇

3. *Esercizio:* Si supponga di dover effettuare un viaggio dalla località A alla località B con un'auto che ha un'autonomia di k chilometri. Lungo il percorso, a partire da A sono presenti n distributori di benzina, ciascuno distante dal precedente meno di k chilometri e l'ultimo dista meno di k chilometri da B . Sia d_i la distanza che separa il distributore i dal distributore $i + 1$, per $i = 1, 2, \dots, n - 1$, sia d_0 la distanza tra A ed il primo distributore e sia d_n la distanza da B dell'ultimo distributore. Descrivere un algoritmo greedy che seleziona un numero minimo di distributori in cui far tappa durante il viaggio. Giustificare le affermazioni fatte.

◇

4. *Esercizio:* Sia dato un intero N , ed un vettore di interi $a = a[1] \dots a[n]$. Il problema è di calcolare il massimo valore di k per cui possiamo trovare indici $1 \leq i_1 < i_2 < \dots < i_k \leq n$ per cui

$$a[i_1] + a[i_2] + \dots + a[i_k] \leq N.$$

Dare un algoritmo greedy che risolve il problema, provandone la correttezza. Come al solito, il primo passo consisterà nell'ordinare il vettore a in modo opportuno.

◇

5. *Esercizio:* Si descriva in dettaglio l'algoritmo **Greedy-Zaino_frazionario** per il problema dello zaino frazionario, e si provi che esso produce una soluzione ottima.

◇

6. *Esercizio:* Si descriva in dettaglio l'algoritmo greedy per il problema della Selezione di Attività, e si provi che esso produce una soluzione ottima.

◇

7. *Esercizio:* Si descriva in dettaglio l'algoritmo greedy per trovare l'ordinamento di lavori con tempo medio di attesa minimo e si provi che esso produce una soluzione ottima.

◇

8. *Esercizio:* Si costruisca un codifica di Huffman per il seguente insieme di caratteri e relative frequenze

a:7 b:7 c:14 d:11 e:10 f:15 i:16 l:16, r:16

Usando la codifica costruita, si codifichi il testo **decifrabile**

◇

9. *Esercizio:* Si codifichi il testo **abbracaddabrrra!!** usando una codifica di Huffman, da calcolarsi in via preliminare sulle relative frequenze dei caratteri così come essi appaiono nel testo.

◇

10. *Esercizio:* Si costruisca un codifica di Huffman per il seguente insieme di caratteri e relative frequenze

a:4 b:5 c:7 d:8 e:10 i:12 r:20.

Usando la codifica costruita, si codifichi il testo **abbededari**

◇

11. *Esercizio:* Dato un alfabeto di caratteri $C = \{c_1, c_2, \dots, c_n\}$, con relative frequenze $f(c_1), f(c_2), \dots, f(c_n)$, si consideri il problema di determinare un albero di codifica ottimo per C . Si dimostri, con precisione, che esiste sicuramente un albero ottimo T in cui i due caratteri $x, y \in C$ di frequenze minime appaiono nell'albero T alla profondità massima, e sono fratelli.

◇

12. *Esercizio:* Si descriva in dettaglio l'algoritmo greedy per il Partizionamento delle Attività, e si provi che esso produce una soluzione ottima.