Esercitazione
31 marzo 2026

programmazione dinamica

problema 1

Esercizio Il signor Valter Bianchi, dopo aver fatto un bel gruzzoletto vendendo cristalli non proprio legali, ha deciso di diversificare la sua attività e si è messo a vendere della roba – anch'essa non proprio legale – che per comodità chiameremo stecca di cioccolata. La stecca di cioccolata può essere venduta tutta intera o può essere spezzata in segmenti più piccoli da vendere separatamente. La lunghezza della stecca di cioccolata è di L centimetri, con L intero. Si assuma che nello spezzare la stecca la lunghezza dei pezzi ottenuti (in centimetri) debba essere ancora un numero intero. Per esempio un pezzo lungo 3 centimetri può essere venduto così, spezzato in tre pezzi da 1 centimetro o in due pezzi: uno da 2 centimetri e l'altro da 1 centimetro (mentre non si possono fare due pezzi da mezzo centimetro e due centimetri e mezzo). Il guadagno che il signor Valter Bianchi riesce a fare se vende un pezzo lungo t centimetri è $G(t)$, $t = 1, 2, \ldots, L$. Progettare un algoritmo di programmazione dinamica che aiuti il signor Valter Bianchi a guadagnare il più possibile. La complessità temporale dell'algoritmo deve essere polinomiale in L.

Un soluzione discussa la trovate qui:

https://www.mat.uniroma2.it/~guala/discussi.pdf

problema 2

Esercizio:

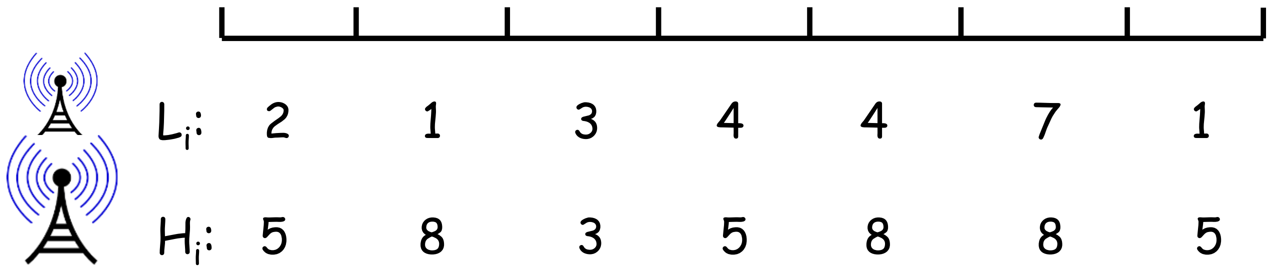
Si vuole dotare una pista ciclabile di un buon servizio wifi. Per fare questo si devono installare dei ripetitori wireless. La pista ciclabile è lunga n tratte. Il costo di installazione di un ripetitore non è uniforme e dipende dalla tratta in cui si installa il ripetitore e dal tipo del ripetitore. In particolare, ci sono due tipi di ripetitori, uno di tipo *high* (H) e uno di tipo *low* (L).

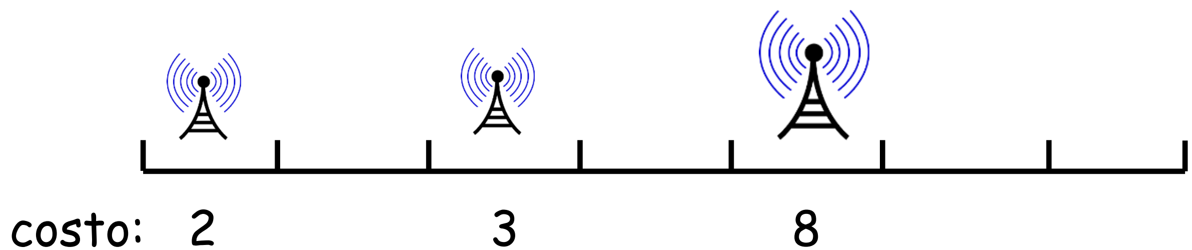
Se nella tratta i si installa:

- un ripetitore di tipo L, il ripetitore è in grado fornire il servizio alle tratte i e $i+1$. Il costo di installazione è L_i
- un ripetitore di tipo H, il ripetitore è in grado fornire il servizio alle tratte i, $i+1$ e $i+2$. Il costo di installazione è $H_i \geq L_i$

Progettare un algoritmo di programmazione dinamica che calcoli una soluzione di costo totale minimo che fornisce il servizio a tutte le tratte.

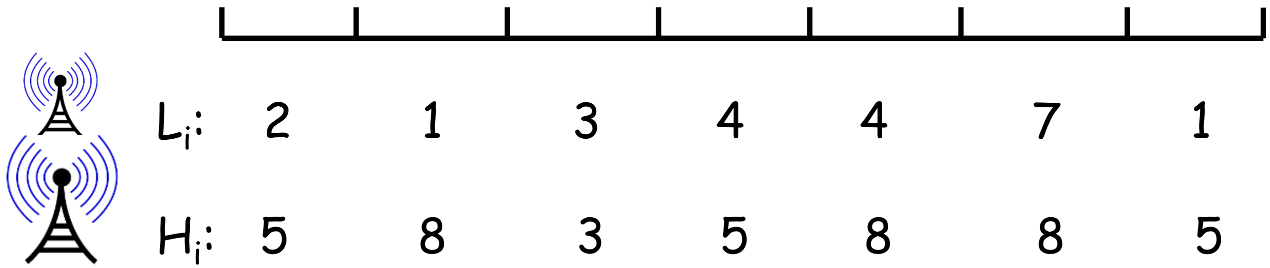
Esempio

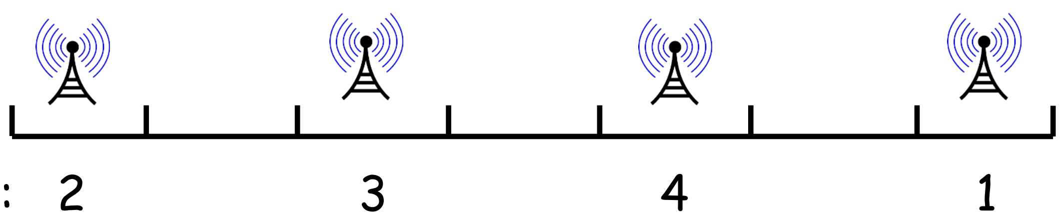




una soluzione di costo totale 13

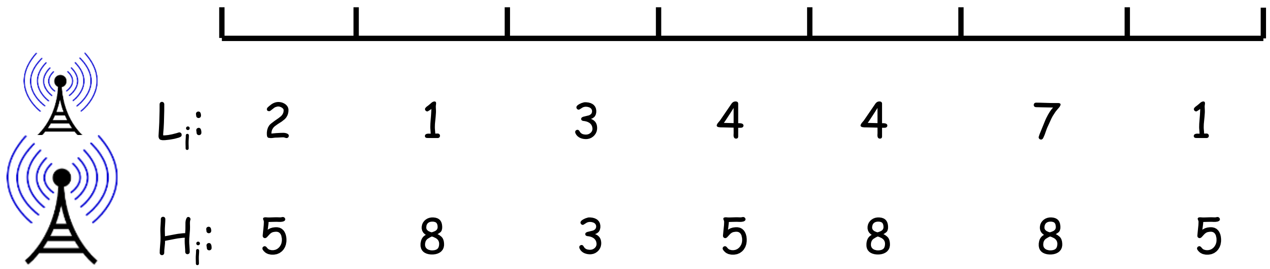
Esempio

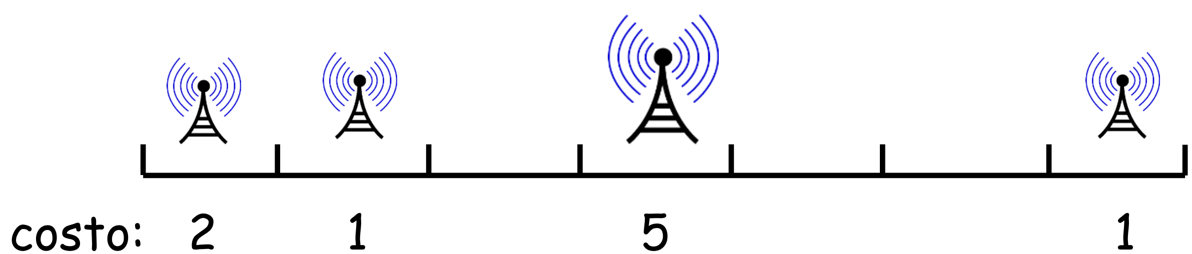




una soluzione di costo totale 10

Esempio

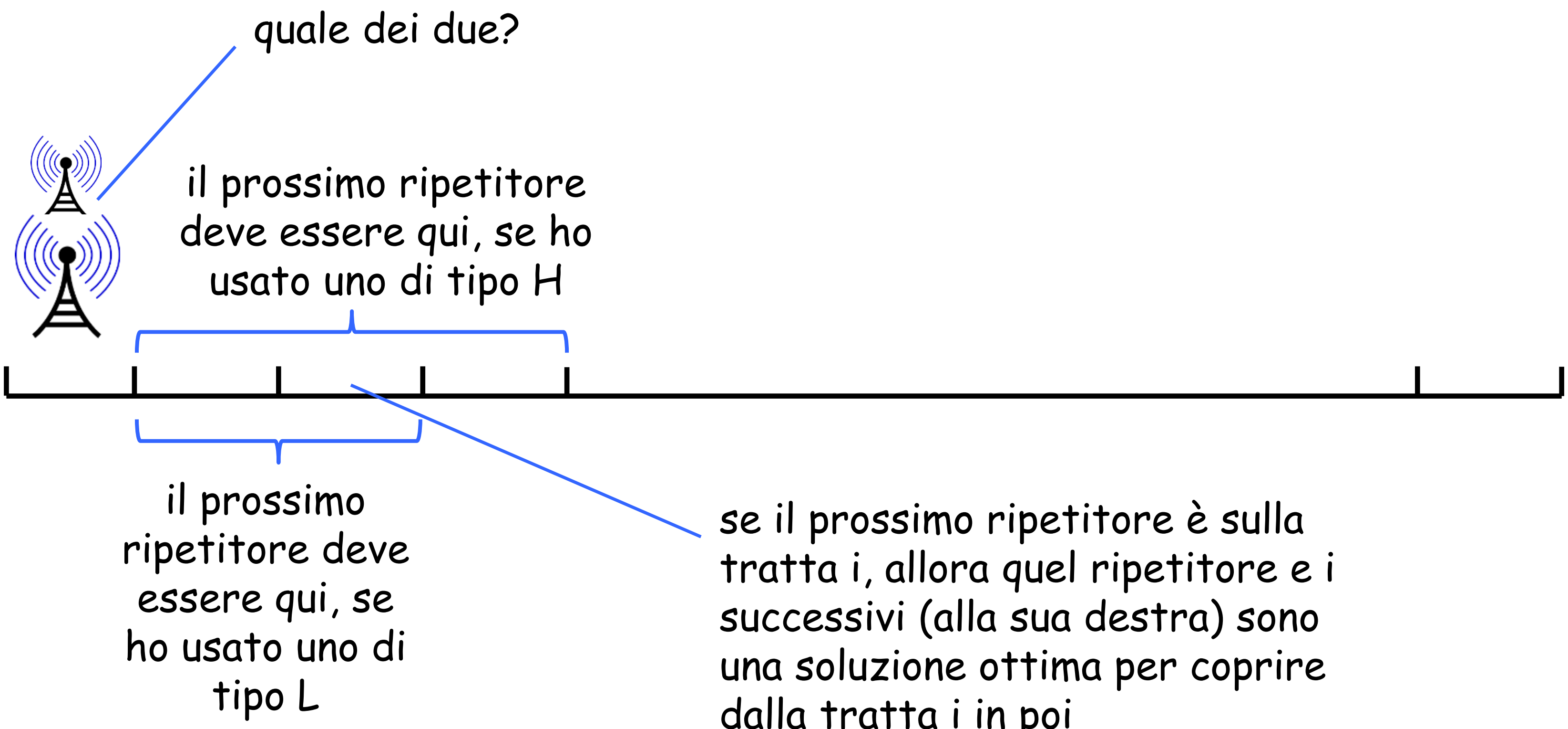




una soluzione di costo totale 9

Idea:

- "indovinare" il tipo di ripetitore installato nella soluzione ottima nella tratta 1
- questo copre alcune tratte che ora non è più necessario coprire
- coprire le tratte "restanti" all'ottimo
- attenzione: non è detto che nell'ottimo non ci sia un ripetitore anche nella tratta 2!



Soluzione

Sottoproblemi:

Opt[i]: costo minimo per coprire le tratte i,i+1,...,n

Soluzione cercata:

Opt[1]

Equazione di Bellman:

$Opt[i]=\min\{L_i+Opt[i+1], L_i+Opt[i+2], H_i+Opt[i+3],\}$

Casi base:

$Opt[i]=0$ se $i>n$

Ordine dei sottoproblemi:

Opt[n], Opt[n-1],..., Opt[2], Opt[1]

pseudocodice

CalcoloCostoInstallazione:

1. Opt[n+1]=Opt[n+2]=0
2. for i=n down to 1 do
    1. Opt[i]= min {L_i+Opt[i+1], L_i+Opt[i+2], H_i+Opt[i+3],}
3. return Opt[1]

running time: O(n)

- come al solito l'algoritmo calcola solo il valore della soluzione ottima
- dato il vettore Opt[] sapete ricostruire in tempo O(n) anche la soluzione?