

Ho provato a mettere insieme alcune cose. Spero possiate capire un po'.  
Comunque non sono cose difficili, io ho molto materiale tra cui molti esercizi  
presi da altri professori che mi son stati utili: se vi servono ve li mando.

## Flusso di costo minimo

**RICERCA**

$C, u,$

**FLUSSO  
COSTO  
MINIMO**

- il problema è  $4 \rightarrow 5 \Rightarrow$  punto da 4 e cerco un'altra strada per arrivare a 5
- Percorso  $4-2-3-5$ 

$$\theta = \min \{ x_{45} - u_{45}, x_{42}, u_{23} - x_{23}, u_{35} - x_{35} \}$$

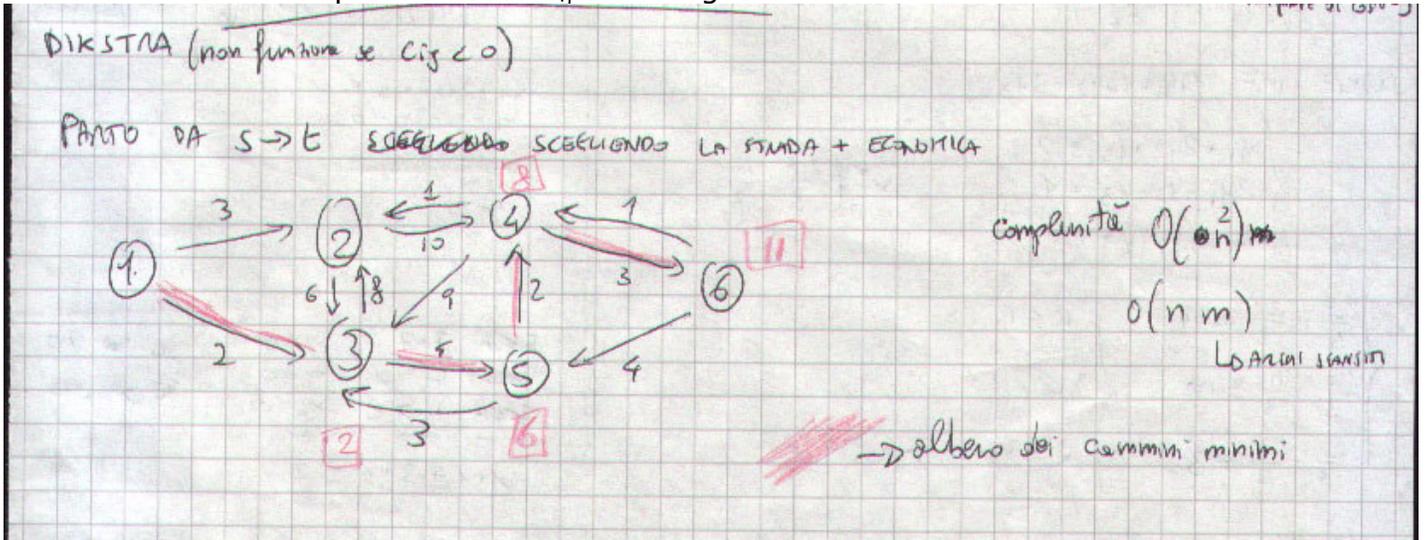
$$= \min \{ 1, 3, 3, 2 \} = 1$$
- aggiungo di 1 il flusso
- nuovi GR

È tipo l'esercizio 9 del compito 2003. Si segnano i cammini aumentanti (cioè dove si può aumentare e dove togliere, disegnando le frecce nelle varie direzioni) poi si calcola il  $\theta$  (con i segni giusti, è spiegato sulle dispense, se no chiedetemi) e quindi dal nodo sorgente (quello con sole frecce uscenti) si invia il flusso che si è calcolato col  $\theta$  scegliendo la strada migliore (costo minimo oppure capacità...) è alla fine si disegna il grafo residuale. Se il grafo residuale presenta un taglio di capacità minima l'esercizio è finito altrimenti significa che esiste un altro cammino aumentante e quindi si ripete da capo ancora tutto. Taglio di capacità minima credo sia definito sulle dispense, altrimenti ditemi che invio un disegno esplicativo.

Nell'esercizio 9 del compito 2003 veniva chiesto se era migliorabile: come si fa? Si vede se esiste un cammino aumentante, si calcola il  $\theta$ , si sceglie il percorso e si disegna il residuale.

## Algoritmo Dijkstra

Non funziona quando i costi  $c_{ij}$  sono negativi.



Partire dal nodo Sorgente (quello che ha sole frecce in uscita) e arrivare al nodo T (generalmente), cioè quel nodo che ha sole frecce in entrata.

Si sceglie per tentativi la strada più economica (nell'esempio 1-3-5-4-6) e quindi si segna nel quadrato sopra il costo totale. Dopo di che quello è l'*albero dei cammini minimi* (quello in rosa nell'esempio). Come si faccia ve lo spiego con un esempio: se guardate l'esercizio 10 della prova dello scorso anno è molto simile (solo che è l'algoritmo SPT\_L): vi fa partire dal nodo radice (in quel caso il 2: dato nel testo dell'esercizio) e dovete toccare una sola volta tutti i cerchi (i nodi) sfruttando la strada minima. Per fare ciò si usa la funzione Q cioè  $Q = \{2, 3, 4, \dots\}$  dove si inserisce il primo nodo in esame (2) si guardano la stella uscente (le frecce che partono da 2) e si valutano i percorsi più economici. Vedo che posso andare verso 3 e 4 quindi metto nella funz. Q anche 3 e 4 e valuto ancora i percorsi uscenti... e così via finché non ho visitato tutti i nodi. (Nei quadrati sopra i nodi si mette la somma delle distanze).

La differenza tra SPT\_L e Dijkstra è sostanzialmente che nel primo potete passare SOLO una volta per ogni nodo (cioè avete solo un tentativo, una volta che avete visitato il nodo passate ai successivi), mentre con Dijkstra dovete continuamente analizzare i nodi (anche più volte) finché non trovate l'albero dei cammini minimi (in poche parole rimettete più volte dentro la Q nodi già visitati e rivisitate).

## Algoritmi Greedy

Se vi chiede un algoritmo greedy non dovete fare nessun conto ma solamente scrivere in italiano come implementereste quell'algoritmo. Non trovo ne mi vengono esempi. Comunque non vi preoccupate, se trovate scritto

“determinare un algoritmo di tipo greedy affinché...” voi gli mettete il vostro ragionamento.

## **Esercizi con variabili**

Vincoli, max-min funzioni, soluzione ottima: sulle dispense c'è tutto. Riscrivere tutto mi viene abbastanza lungo. Comunque si determina la soluzione ottima per max o min qualcosa e si devono mettere i vincoli in base all'esercizio. Anche questo è abbastanza semplice, basta fare molti esercizi.