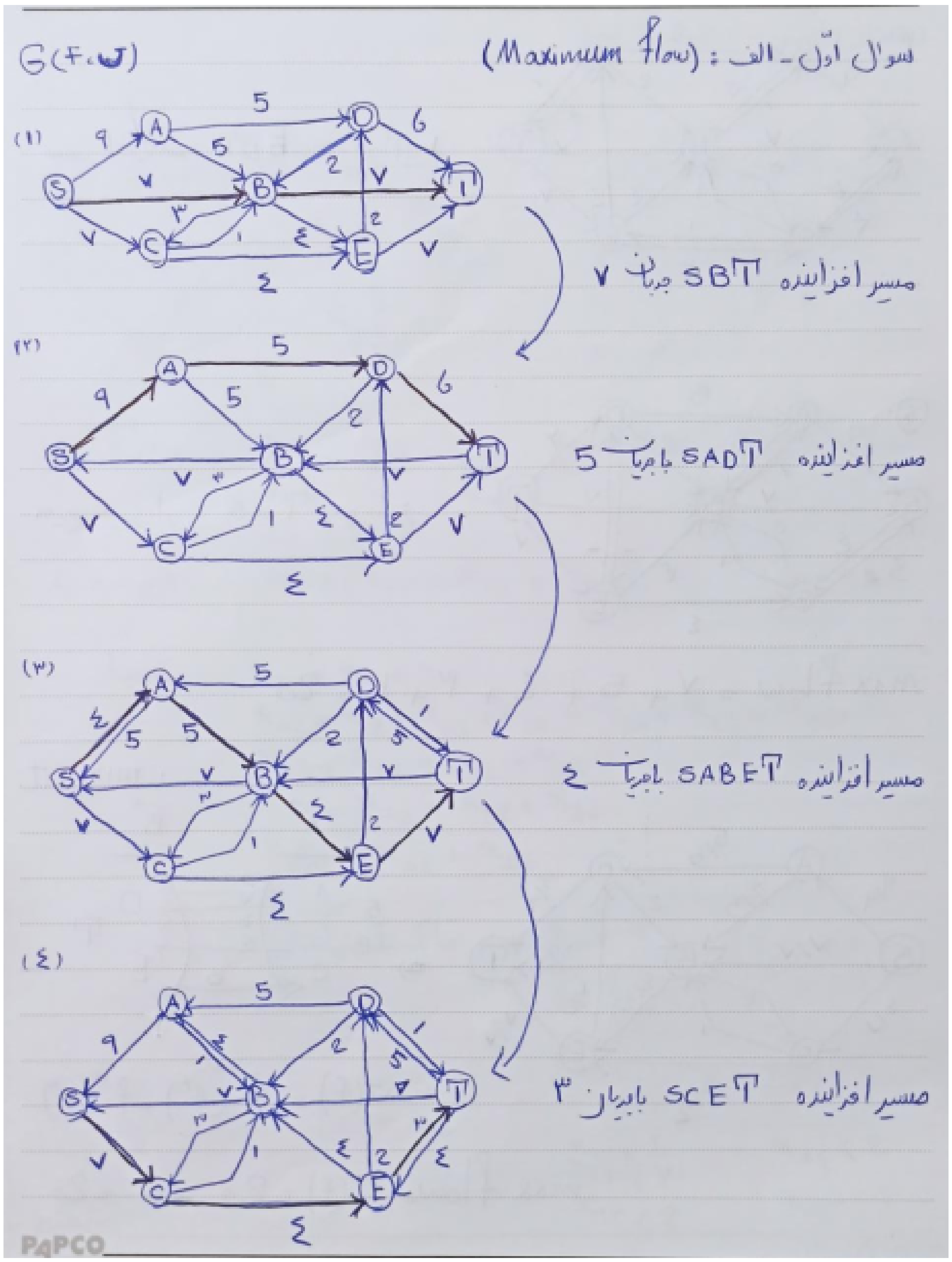
بسم الله الرحمن الرحیم

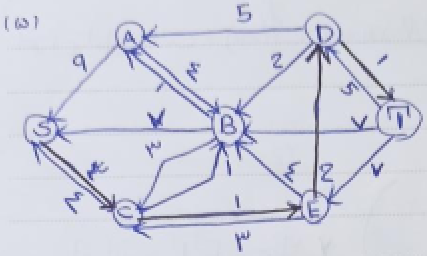
پاسخ تکلیف سری اول
الگوریتم‌های پیشرفته
پاییز ۱۴۰۰

پاسخ سوال اول)

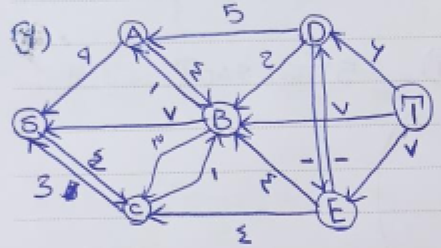
قسمت الف)



ادامه‌ی پاسخ قسمت الف سوال اول)



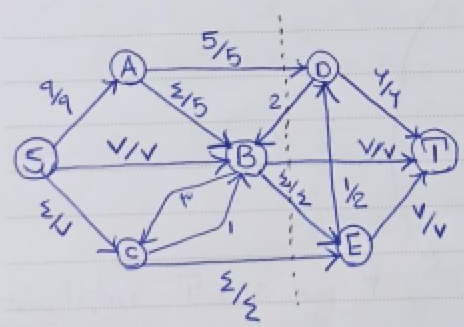
مسیر افزاینده SCEDT جریان ۱

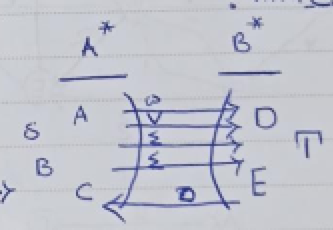


هیچ مسیری از S به T وجود ندارد

$\max flow = ۷ + 5 + ۴ + ۳ + ۱ = 20$

: min cut

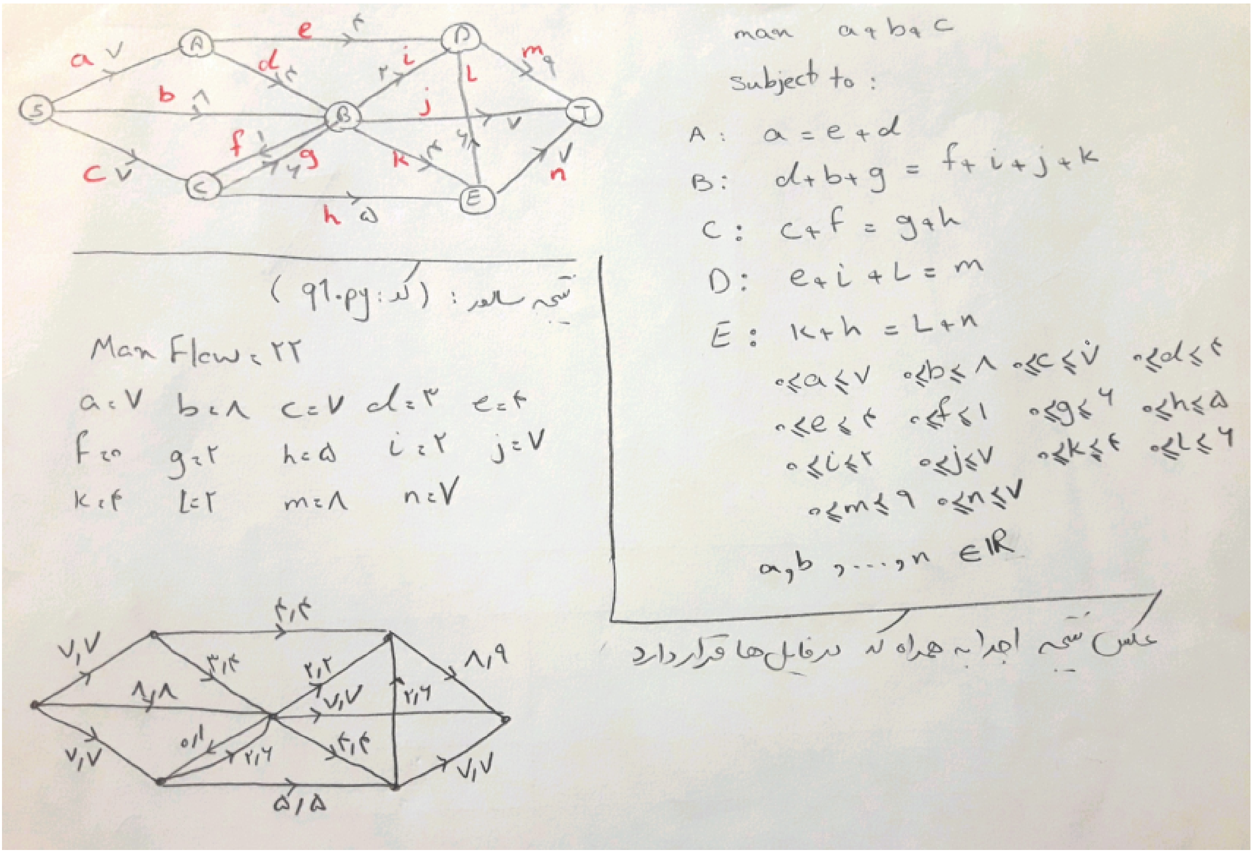




$v(f) = f^{out}(A^*) - f^{in}(A^*)$

$\max flow = v(f) = 20 - 0 = 20$

قسمت ب)



$\max \; a+b+c$

subject to :

A : $a = e + d$

B : $d + b + g = f + i + j + k$

C : $c + f = g + h$

D : $e + i + L = m$

E : $k + h = L + n$

$0 \le a \le 7 \quad 0 \le b \le 8 \quad 0 \le c \le 7 \quad 0 \le d \le 4$

$0 \le e \le 4 \quad 0 \le f \le 1 \quad 0 \le g \le 6 \quad 0 \le h \le 5$

$0 \le i \le 2 \quad 0 \le j \le 7 \quad 0 \le k \le 4 \quad 0 \le L \le 6$

$0 \le m \le 9 \quad 0 \le n \le 7$

$a, b, \dots, n \in \mathbb{R}$

نتیجه سالور : (کد: ۹۱.py)

Max Flow = ۲۲

$a=7 \quad b=8 \quad c=7 \quad d=3 \quad e=4$

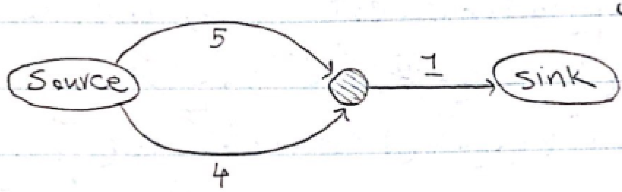
$f=0 \quad g=2 \quad h=5 \quad i=2 \quad j=7$

$k=4 \quad L=2 \quad m=8 \quad n=7$

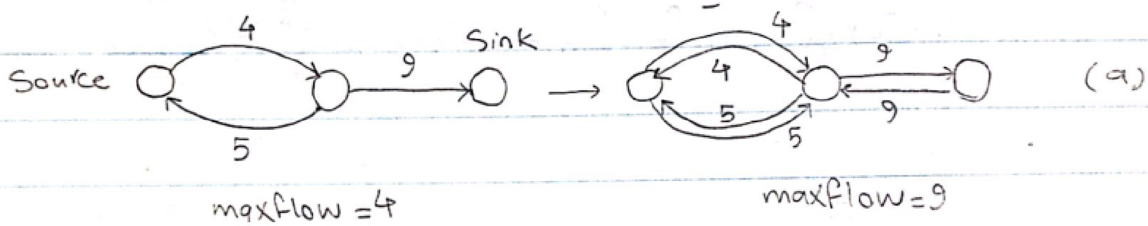
عکس نتیجه اجرا به همراه کد در فایل‌ها قرار دارد

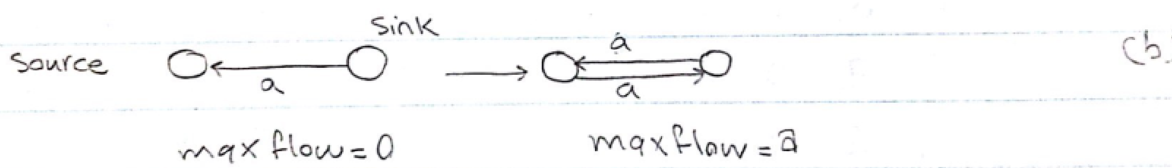
پاسخ سوال دوم)

الف: غلط، برای مثال در گراف زیر با این وجود که ظرفیت همه یال‌ها متمایز است، اما در نهایت شار بیشینه به خاطر محدودیت ظرفیت یال ورودی به رأس sink همیشه برابر 1 است، اما از مسیرهای متفاوت



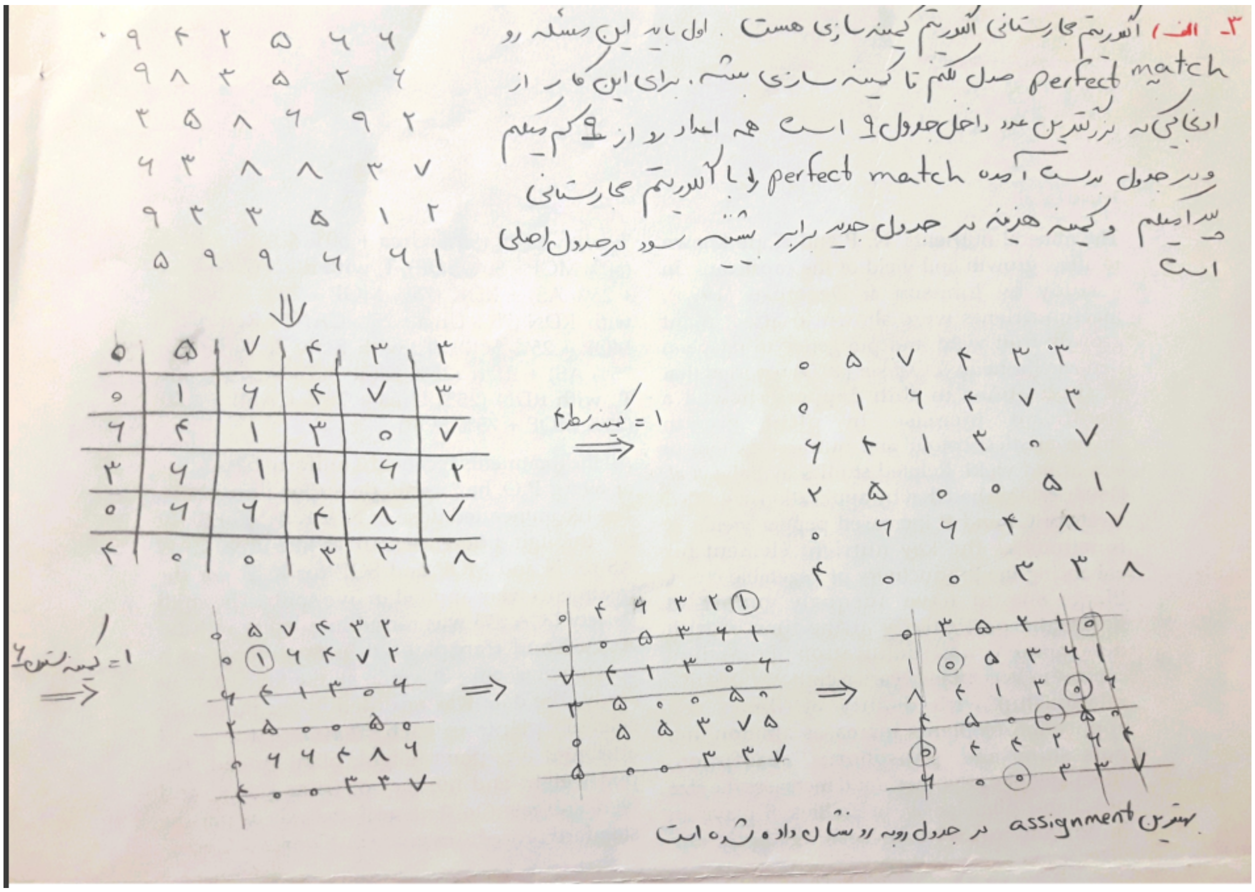
ب: غلط: مثال نقض شکل زیر،





پاسخ سوال سوم)

قسمت الف)



قسمت ب)

ب) اسم ماتریس بالا را A میگذاریم

۳۶ متغیر x_{ij} داریم. که یک بودن آن به معنای سپردن کار i به کارگر j است

$i, j \in \{0, 1, 2, \dots, 5\}$

$$\max \sum_{i=0}^{5} \sum_{j=0}^{5} A_{ij} x_{ij}$$

$$\sum_{i=0}^{5} x_{ij} = 1 \quad \text{for } \forall j \rightarrow$$ هر کارگر فقط یک کار

$$\sum_{j=0}^{5} x_{ij} = 1 \quad \text{for } \forall i \rightarrow$$ هر کار فقط یک کارگر

$$0 \leqslant x_{ij} \leqslant 1 \quad \text{for } \forall i, j$$

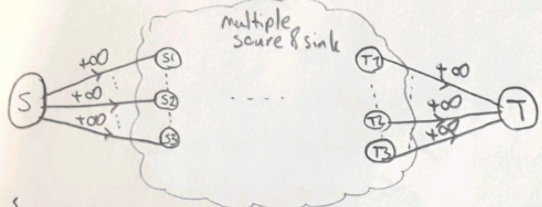
نتیجه سالور و کد در کنار این فایل قرار دارد

(q3.py)

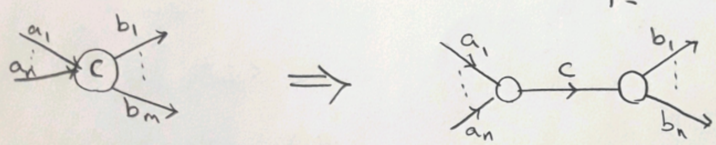
پاسخ سوال چهارم)

قسمت الف)

برای مدل اول یک source و sink جدید به شکل روبه‌رو ایجاد کنیم و از آن به همه source و sink های موجود با ظرفیت بی‌نهایت یال بکشیم. و مسئله جدید را حل کنیم



برای مدل دوم هر رأس را دوتا می‌کنیم که ورودی‌ها به رأس اول بروند و خروجی‌ها از رأس دوم خارج شد و از رأس اول به رأس دوم یک یال با ظرفیت رأس می‌دهیم



قسمت ب)

ب)

برای مدل سوم دقیقاً مانند حالت اصلی در نظر می‌گیریم x_{ij} شار عبوری از راس i به j باشد (از روی یال i به j) و u_{ij} کران بالا ظرفیت یال ij و l_{ij} ظرفیت کران پایین آن باشد

maximize $\sum x_{1j} \rightarrow$ جریان خارج شده از s

subject to $\sum x_{ji} - \sum ij = 0 \rightarrow$ برای هر راس جریان ورودی و خروجی برابر باشد

$l_{ij} \leq x_{ij} \leq u_{ij} \rightarrow$ جریان روی هر یال بین کران بالا و پایین باشد

برای مدل چهارم اولاً شار ورودی به T باید max شود تا اتلاف ها نیز در نظر گرفته شود. ثانیاً رابطه جریان ورودی و خروجی هر راس تغییر می‌کند (α_i ضریب اتلاف راس i است)

maximize $\sum x_{iT} \rightarrow$ جریان ورودی به T (راس سینک)

subject to $(1-\alpha_i)\sum_j x_{ji} = \sum_j x_{ij}$ برای هر راس i، α_i برابر ورودی ها اتلاف شود

$0 \leq x_{ij} \leq u_{ij}$

پاسخ سوال پنجم)

There is a polynomial-time algorithm to determine the largest *k* for which a graph *G* is *k*-edge-connected. A simple algorithm would, for every pair *(u,v)*, determine the maximum flow from *u* to *v* with the capacity of all edges in *G* set to 1 for both directions. A graph is *k*-edge-connected if and only if the maximum flow from *u* to *v* is at least *k* for any pair *(u,v)*, so *k* is the least *u*-*v*-flow among all *(u,v)*.

If *n* is the number of vertices in the graph, this simple algorithm would perform $O(n^2)$ iterations of the Maximum flow problem, which can be solved in $O(n^3)$ time. Hence the complexity of the simple algorithm described above is $O(n^5)$ in total.

An improved algorithm will solve the maximum flow problem for every pair *(u,v)* where *u* is arbitrarily fixed while *v* varies over all vertices. This reduces the complexity to $O(n^4)$ and is sound since, if a cut of capacity less than *k* exists, it is bound to separate *u* from some other vertex. It can be further improved by an algorithm of Gabow that runs in worst case $O(n^3)$ time. [4]

The Karger–Stein variant of Karger's algorithm provides a faster randomized algorithm for determining the connectivity, with expected runtime $O(n^2 \log^3 n)$.[5]

A related problem: finding the minimum *k*-edge-connected spanning subgraph of *G* (that is: select as few as possible edges in *G* that your selection is *k*-edge-connected) is NP-hard for $k \geq 2$.[6]

پاسخ سوال ششم)

کاهش مسئله با LP :

متغیر تصمیم x^{k}_{vu} به معنای جریان عبوری از نوع k و از رأس v به رأس u می باشد $(v,u)\in E$

و C_{vu} همان ظرفیت یال (v,u) می باشد

$$\text{maximize} \quad \sum_{i=1}^{k} \sum_{v:(s_i,v)\in E} x^{i}_{s_i v}$$

Subject to :

$$\forall_{1\le i\le k} \quad \sum_{v:(s_i,v)\in E} x^{i}_{s_i v} = \sum_{v:(v,t_i)\in E} x^{i}_{v t_i}$$

$$\forall_{1\le i\le k} \quad \sum_{v:(s_i,v)\in E} x^{i}_{s_i v} \geq d^{k}$$

$$\forall_{1\le i\le k} \quad \forall v\in V \quad \sum_{u:(u,v)\in E} x^{k}_{uv} = \sum_{u:(v,u)\in E} x^{k}_{vu}$$

$$\forall_{1\le i\le k} \quad \sum_{j=1,\, j\neq i}^{k} \sum_{v:(s_i,v)\in E} x^{j}_{s_i v} = 0$$

$$\forall_{1\le i\le k} \quad \sum_{j=1,\, j\neq i}^{k} \sum_{v:(v,t_i)\in E} x^{j}_{v t_i} = 0$$

$$\forall (v,u)\in E \quad \sum_{i=1}^{k} x^{i}_{vu} \leq C_{vu}$$

$$\forall_{1\le i\le k},\ \forall (v,u)\in E \quad x^{i}_{vu} \geq 0$$