۱.

(آ) نادرست. طبق صفحه ۳۷ سورس، این عامل باید اکشنی که انتظار (expect) می‌رود ماکسیمم performance measure را دارد، انتخاب کند و شروط ذکر شده کافی نیست.

(ب) درست. طبق صفحه 37 سورس، عقلانی بودن به 4 عامل وابسته است که عامل دوم prior knowledge of the environment است.

(ج) درست.

از آنجا که بودن sideway ها با احتمال $p = 0{,}14$ به نتیجه می‌رسد پس طبق توزیع هندسی امید ریاضی تعداد مراحل مورد نیاز $\frac{1}{p} = \frac{100}{14} \approx 7$ است.

$$\rightarrow \underset{\text{avg failure}}{6 \times 3} + \underset{\text{avg success}}{4} = 22$$

اما با احتساب sideway ها: احتمال $= 0{,}94$

$$\frac{1}{0{,}94} \approx 1{,}06 \text{ iteration}$$

$$\underset{\text{avg success}}{(1 \times 21)} + \left(\frac{0{,}06}{0{,}94}\right) \times \underset{\text{avg failure}}{64} \approx 25$$

$$\boxed{25 > 22}$$

* به دلیل کم بودن تعداد local maxima ها و plateaux ها برای استارت رندوم در هیل کلایمینگ این مسئله بهینه است.

(د) درست. طبق اثبات پاراگراف دوم صفحه 85 سورس

(ه) درست. چون تنها مرحله آخر مقدار t را ذخیره می‌کند و همچنین به دلیل کاربری حرکت رندوم آ به سان mutation در genetic می‌تواند تا حدی و نه لزوماً از بهینه‌های محلی فرار کند.

-۲

(آ) حالت ها تمام جایگیری های این n کارشناس در صفحه $n\times n$ است. ← $n\times n^2=n^3$ تا که چون نباید برخورد داشته باشند، $\binom{n^2}{n}$ است. کنش ها تمام انتخاب های حرکت های n کارشناس به ↓ ← ↑ → و ثابت است که کل 5^n است که تنها حالت های عدم تداخل و داخل جدول محسوب می شوند.

حالت شروع n تایی $((1,1),(2,1),\dots,(n,1))$ است و هدف $((n,1),(n-1,1),\dots,(1,1))$

هزینه هر کنش جمع هزینه هر کنشه است که در واقع می شود تعداد حرکات غیر ثابت کارشناس ها.

ب) n^2 جایگاه برای n شئ ← $\underbrace{n^{n^2}}$ البته دقیق تر $\binom{n^2}{n}$ است که باز $O(n^{n^2})$ است.

$$\log\binom{n^2}{n} = \log(n^2!) - 2\log(n!) = O(n^2\log(n^2) - 2n\log(n)) = O(2n^2\log(n) - 2n\log(n)) = O(n^2\log n)$$

$\binom{n^2}{n} = \frac{n^2!}{n!\,n!}$

$\log(n!) = O(n\log n)$

$\binom{n^2}{n} = O(2^{n^2\log n}) = O(n^{n^2})$

ج) در هر حالت هر کارشناس حداکثر 5 کنش دارد ← $d = 5^n$ ← # کارشناس ها

د) هیوریستیک را برابر فاصله منهتن مکان کنونی و مکان هدف کارشناس در نظر می گیریم.
چون در هر حرکت حداکثر یکی به مختصات x یا y اضافه یا کم می شود و حداکثر یکی از آنها هم تغییر می کند
پس تعداد حرکات لازم حداقل فاصله منهتن است ← $h_1^*(x) \geq h_1(x)$ ⇐ قابل قبول است.
در صورت وجود کارشناس های دیگر هم چون حرکات آنها صرفاً در حرکت کارشناس خاص مد نظر محدودیت بیشتری قرار می دهد.
و این حالت هم قابل قبول است ⇐ $h^*_2(x) \geq h_1^*(x) \geq h_1(x)$

(ه) * از آنجا که هر یک از کارشناس ها به حداقل $h(x)$ ارائه شده در بخش د برای رسیدن به مقصد نیاز دارند،
پس برای هیوریستیک کلی

$$\forall i\in\{1,\dots,n\}\; h^*(x_1,\dots,x_n) \geq h_i(x_i)$$

$$\Rightarrow h^*(x_1,\dots,x_n) \geq \max(h_i) \rightarrow \text{قابل قبول است}$$

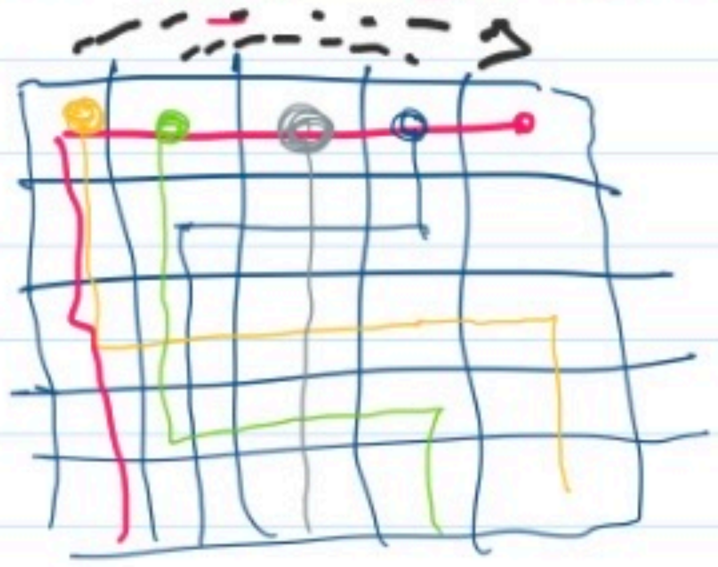
حال داریم که:

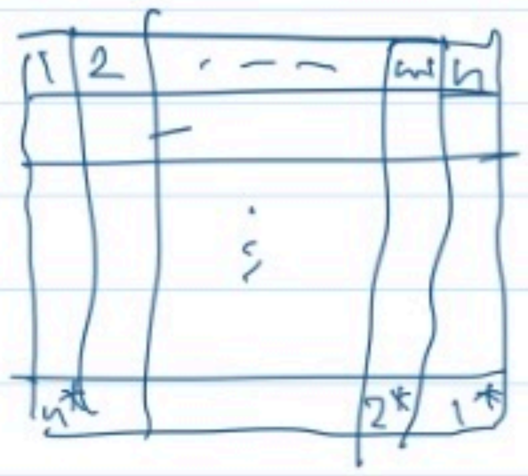
$$\min(h_1,\dots,h_n) \leq \max(h_1,\dots,h_n)$$

$$h_i \leq \max(h_j) \Rightarrow h_1+\dots+h_n \leq n\max(h_i) \Rightarrow \frac{\sum h_i}{n} \leq \max(h_i)$$

$\Rightarrow \min(h_1,\dots,h_n),\ \max(h_1,\dots,h_n),\ \frac{\sum h_i}{n} \rightarrow$ قابل تبول اند.

حال ثابت می‌کنیم حلقه وجود دارد که مثال نقض سایرین است.





مثال‌ها برای n بزرگتر

در این صورت دقیقاً $\max(h_i) =$ ماکسیمم فاصله منهتن‌ها $= h^*$

چون هر مرحله آن‌ها که به مقصد نرسیدن دقیقاً ۱ حرکت در راستای کاهش فاصله منهتن دارند.

$h^* = 2(n-1)$ (وسطی) $\min(h_i) = n-1$ (فرد) یا n (زوج) در مرحله اول

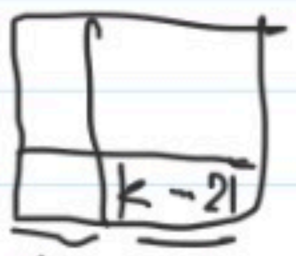
$\sum h_i \geq n\min(h_i) = (n(n-1) \wedge n^2) > 2(n-1)$ مگر n کوچک‌تر ۲ ها

$n\max(h_i) = n(2(n-1)) = 2n(n-1) > 2(n-1) = h^*$ → مگر $n=1$

$\Rightarrow$ $n\min(h_i)$, $\sum h_i$, $n\max(h_i)$ → قابل قبول نیستند

(و) حالات همه جایگیری‌های n کاوشگر اند بجز حالاتی که سطر آخر از انتها تا جایی پر شده باشد

یعنی یا سطر آخر خالی و یا به صورت



پر / خالی

باشد. در این صورت به قطع قطعی در دسترس شود

اما امکان رفت و بازگشت به خانه هدف است پس تعریف سبز را برداشت کردم.

گفته اول به مقصد برسد نه لزوماً بماند.

در این صورت یک لیست n تایی داریم که وقتی i به جایگاه مقصدش برسد (اولین بار) آن ☑ می‌خورد.

حال ← اگر خالی باید این صورت باشد، حالاتی که توالی از اسم k در وضعیت خالی نباشد را حالات ($n \geq k \geq 1$) مسئله به حساب نمی‌آوریم، نمی‌توان به آنها رفت. (یک نوع حافظه است اما با بی‌حافظگی مدل در تضاد نیست)

۲- حالات همه جایگیری‌های n کاوشگر در صفحه $n\times n$، و همه حالات جدول $1\times n$ گرفته شده که هر خانه چه حالت.

$\Rightarrow n^{n^2}\times 2^n \xrightarrow[\text{عدم همسانی}]{\text{دقیق‌تر}} \binom{n^2}{n}\times 2^n \xrightarrow[\text{ضعیف تخمین زدن}]{} O(n^{n^2}\times 2^n)$

کنش ها : حرکت ۱ کاوشگر به یک خانه مجاور درون جدول و ممکن است تیک خوردن یک خانه

یا ثابت ماندن

از جدول $n \times n$ جدا یا ن برای آن کاوشگر باشد. (به شرط آنکه جدول $\boxed{\checkmark}$ برای اما بماند دیگر کنش نمی‌توان کرد.)

مکرر ثابت

برای حالت شروعی که هنوز تا سر جا خورده انه.

هزینه هر کنش : هر حرکت غیر ثابت هزینه ۱ دارد.

هدف : مشابه جایگیری کاور هدف بخش قبل و چک لیست تمام تیک.

در مسئله ها نیازی به بررسی نیست

با تجربه ترتیب کنش قطعاً معادل اولویت رسیدن است.

ب - اندازه فضا

$$\text{اندازه فضا} \lessapprox \binom{n^2}{n} \times 2^n = O\left(n^{n^2} \times 2^n\right)$$

خانه ها — جداول

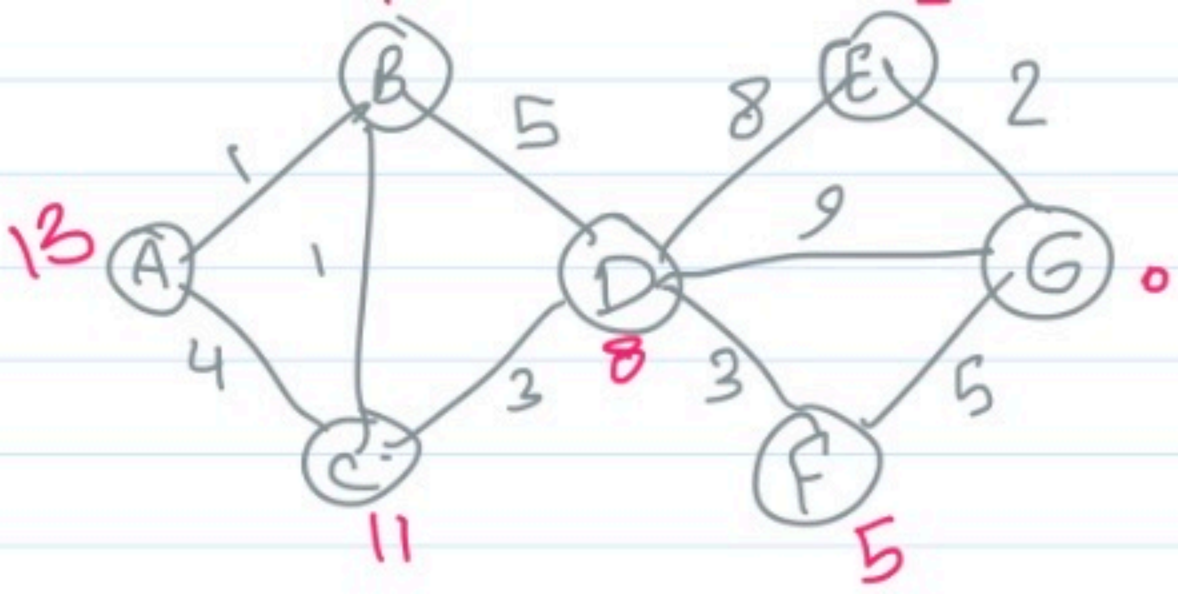
بخش ب خود سوال

این را به دست آوردیم

ج) در بهترین حالت هم تنها یک کاوشگر را می‌توان جابجا یا ثابت گذاشت. دقت می‌کنیم که تیک خوردن نتیجه قرارگیری در حالت است و انتخاب با ما برای تیک زدن یا فقط نزدن نیست که در حالت نشود.

$\rightarrow \quad b = 5 \quad \rightarrow$ چون تنها یک کاوشگر است.

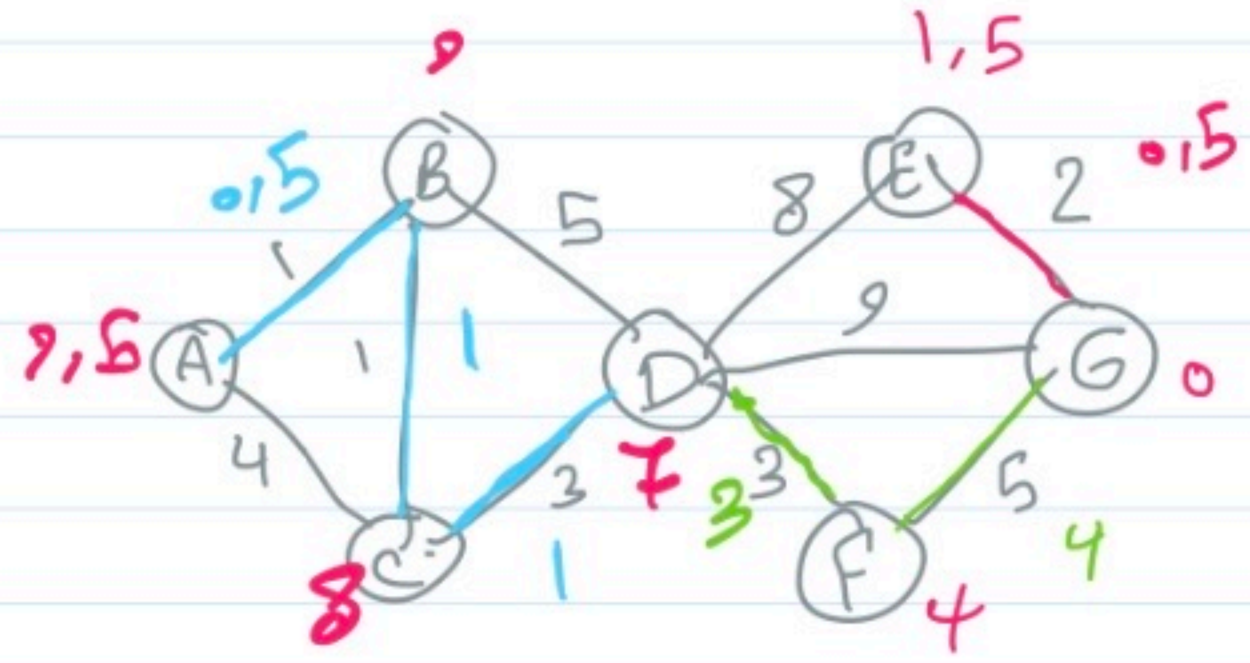
-۳



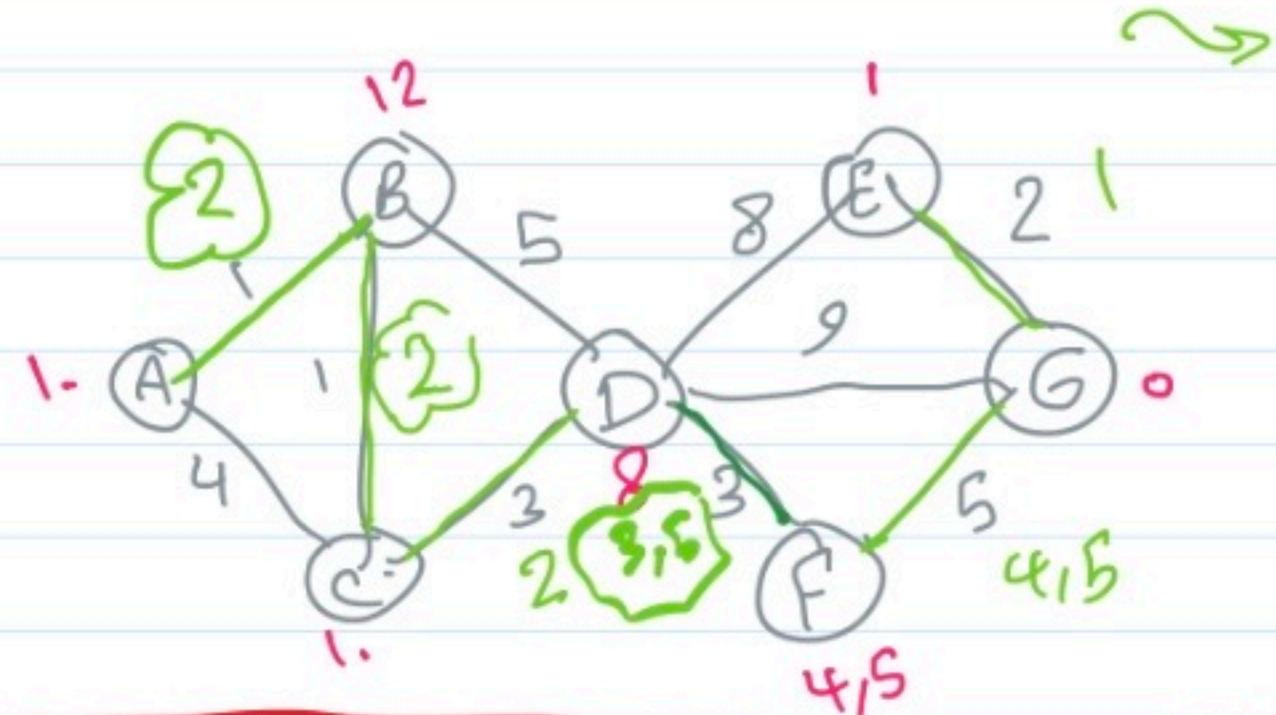
آ)

	h^*	h_1	h_2
A	13	9,5	10
B	12	9	12
C	11	8	10
D	8	7	8
E	2	1,5	1
F	5	4	4,5
G	0	0	0

=> $h_2(x) \le h^*(x)$
$h_1(x) \le h^*(x)$ → h_1, h_2 قابل قبول اند.



$\Delta h(i,j) \le C(i,j)$
=> h_1 یکنواست.

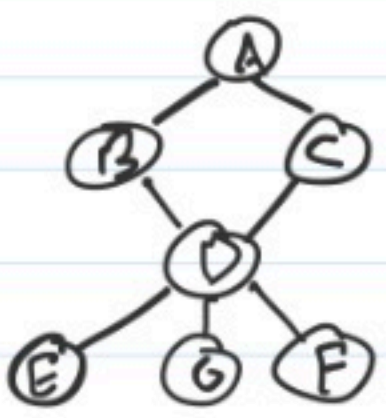


$2 > 1$ => h_2 یکنوا نیست.

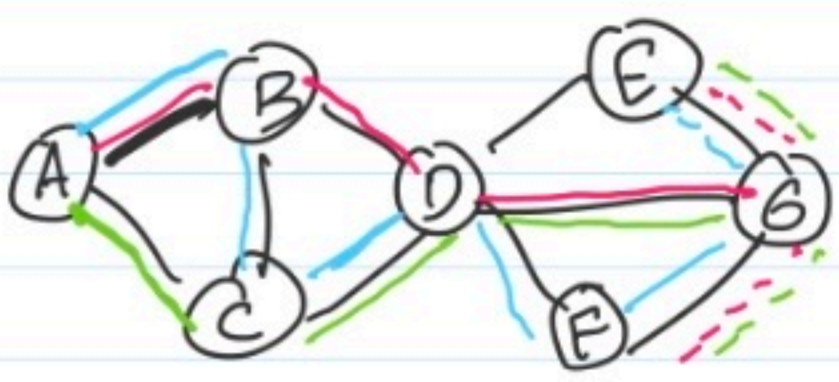
ب)

BFS سطح اول → 4 ← کمترین تعداد یال ← متوقف

A-B-D-G ✓ A-C-D-G ✓ A-B-C-D-F-G ✗



DFS عمق اول



بسته به نوع ترتیب‌دهی هر کدام می‌شود.

A-B-D-G ✓ A-C-D-G ✓ A-B-C-D-F-G ✓

جست و جوی هزینه یکنواخت → یال منفی نداریم ⇐ optimal است.

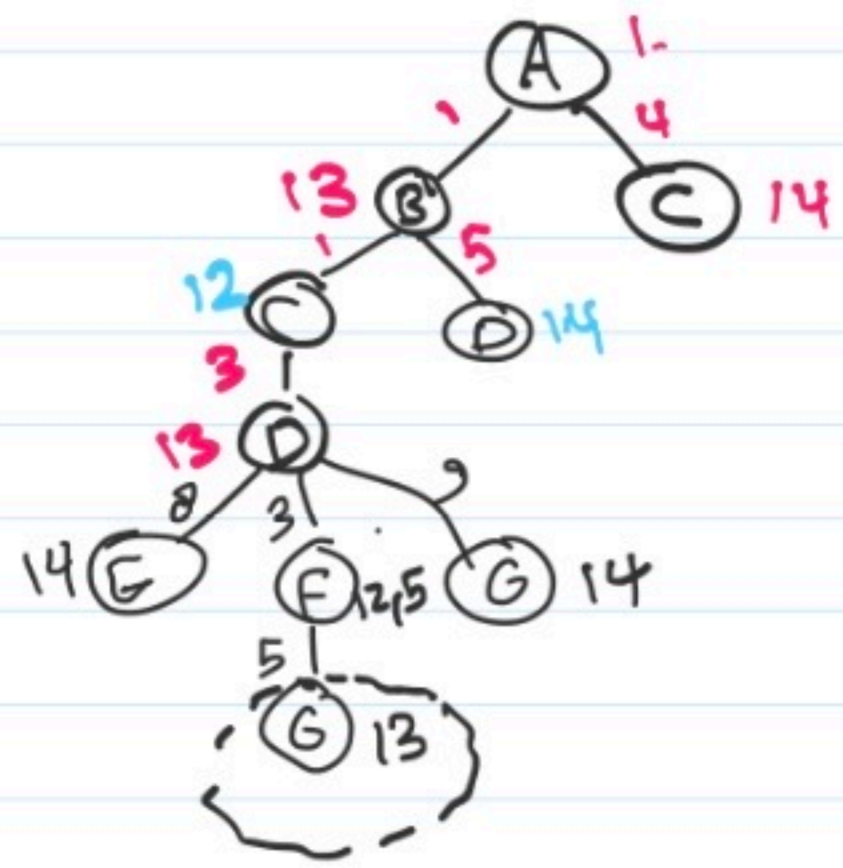
A-B-D-G (15) ✗ A-C-D-G (16) ✗ A-B-C-D-F-G (13) ✓

جست و جوی A^* با h_1
جست و جوی A^* با h_2
درختی (با رأس تکراری) → قابل قبول پس optimal اند → A-B-D-G ✗ A-C-D-G ✗ A-B-C-D-F-G ✓

گراف سرچ

جست و جوی A^* با h_1 → h_1 یکنوا است → optimal است. → A-B-D-G ✗ A-C-D-G ✗ A-B-C-D-F-G ✓

جست و جوی A^* با h_2 → h_2 یکنوا نیست → لزوماً optimal نیست → A-B-D-G ✗ A-C-D-G ✗ A-B-C-D-F-G ✓



closed	open
	A
A	B,C
A,B	C,D
A,B,C	D
A,B,C,D	E,F,G
A,B,C,D,F	E,G

⇒ A, B, C, D, F, G

* ولی اگر از B آغاز داشتیم چون یکنوا نبود ممکن بود به A برویم و مسیر غلط بشود چون در مسیر درست A-B-C-... $h(B)>h(A)$ رخ داده است.

ج)

	h^*	h_3
A	13	10
B	12	?
C	11	9
D	8	7
E	2	1,5
F	5	4,5
G	0	0

$h(x)\le h^*(x) \Rightarrow \boxed{h_3(B)\le 12}$

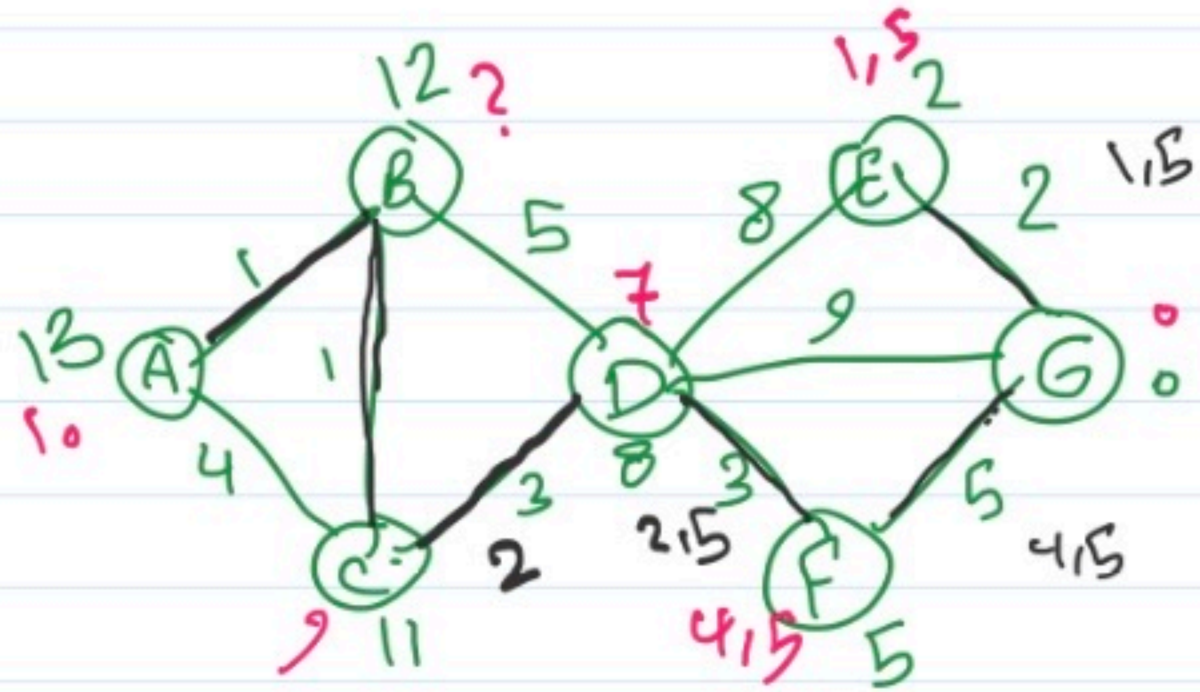
مابقی درست است.

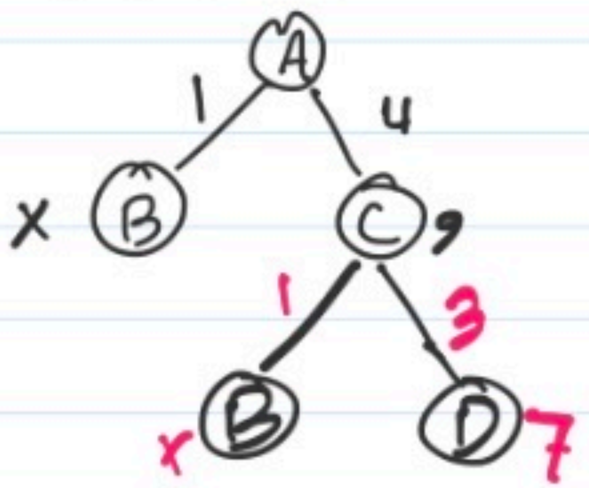
البته منطقاً چون B هدف نیست $h_3(B)\neq 0$ باید گرفت. ($0\le h_3(B)$ کلاً چون یالها نامنفی اند)

$* \; 9 \le x \le 10$

$9 \le x \Leftarrow 10 - x \le 1$

$x \le 10 \Leftarrow x - 9 \le 1$





۵) C سپس D سپس $\Leftarrow f(B) \ge f(C)$

$9+4 \le x+1 \Rightarrow x \ge 12$

$\Leftarrow f(D) \ge f(B) \Leftarrow$ D سپس B

$4+3+7 \ge \min(5+x, x+1) = x+1 \Rightarrow 13 \ge x$

، چون A, B, C closed اند $\xrightarrow{\text{سپس}}$ D $\Leftarrow$ $\boxed{13 \ge x \ge 12}$

4) آ) تمام جایگشت های این n کلمه می‌شود.

درصورت تمایز $n!$

غیرتمایز $x_1, \dots, x_k$ تا باشند که $\sum_{i=1}^{k} x_i = n$

$$\frac{n!}{x_1! \cdots x_k!}$$

* تابع هدف مثل تعداد کلمات غیر جا خود

ب) دو جمله که فقط دو کلمه آن با یکدیگر جابجا شده اند را مجاور می‌نامیم.

هرچه زودتر خواهش می‌کنم آزاد کنید من را

هرچه آزاد خواهش می‌کنم زودتر کنید من را

هرچه زودتر می‌کنم خواهش آزاد کنید من را

درصورت تمایز لغات $\binom{n}{2}$ حالت مجاور دارد که کلمات جابجا شوند.

ج) خیر، ممکن است به حالتی برسد که تغییر هر دو جایگاه کلمات از معنا بکاهد اما همچنان جمله معنا کامل به قطر را نده‌د. (که بسته به نوع تابع تعیین معنادار بودن است).

درصورت اعمال رندوم ولی می‌تواند از این حالات خارج شود مثل جابجایی رندوم در دایره که شاید سپس می‌رسی بهتر که معنادارتر باشد و در مسیر رسیدن به معنای اصلی

و بین کاهش ندارد فقط حداکثر (حداکثر ۲) افزایش است.

* اما در تابع ما حالت بن‌بست وجود ندارد چون اگر کلمه‌ای در جایش نیست و به جایش برود حالت بهتر شده (کلمه جایش قطعاً در جای اصلی خود نیست)

د) طبق اعداد به دست آمده از معنا پذیری های حالات مجاور به نسبت احتمال با آنها n تای بعدی را به دست آورده و سپس یک حد مرز برای جابجایی بین کلمات یعنی اگر k کلمه $k-1$ حالت رندوم انتخاب شده سپس دو جمله کلمات متوالی (در ساخت) این بخش مثل مرزشان جابجا می‌شود. ← cross over

در انتها هم به عنوان mutation می‌توان، یک بازه دایره رندوم را گزینش کرده و آنها را با هم جابجا کرد یا (در هر جمله) اعمال چون آوردن دایره جایگاه رندوم به جایگاه اول.

(5

(آ

$$X = (x_1, \dots, x_n)$$

$$f = \|x\|_2^2 = x_1^2 + \dots + x_n^2$$

$$\frac{\partial f}{\partial x_i} = \frac{\partial\, x_1^2 + \dots + x_n^2}{\partial x_i} = 0 + 2x_i + 0 \Rightarrow \nabla f = (2x_1, \dots, 2x_n) = 2X$$

$$Ax = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + \dots + a_{1n}x_n \\ \vdots \\ a_{n1}x_1 + \dots + a_{nn}x_n \end{pmatrix}$$

$$\Rightarrow f = \|Ax\|_2^2 = \sum_{i=1}^n \Big(\sum_{j=1}^n a_{ij}x_j\Big)^2 \rightsquigarrow \frac{\partial f}{\partial x_t} = \sum_{i=1}^n \Big(2\sum_{j=1}^n a_{ij}x_j\Big)(0 + a_{it} + 0) =$$

$$\left(2\sum_{i=1}^n a_{it}\Big(\sum_{j=1}^n a_{ij}x_j\Big)\right)_{t=1,\dots,n} = 2A^TAx$$

$$2\begin{pmatrix} a_{11} \dots a_{1n} \\ \vdots \\ a_{n1} \; a_{nn} \end{pmatrix}\begin{pmatrix} \sum_{j=1}^n a_{ij}x_j \end{pmatrix} = \begin{pmatrix} \sum a_{i1} \sum_{j=1}^n a_{ij}x_j \end{pmatrix}$$

روش دوم : $\|X\|_2^2 = tr(x^Tx)$

$$\Rightarrow f = \|m\|_2^2 \Rightarrow df = tr(2m^T dm) \,,\; \|Ax\|_2^2 = tr(2x^TA^T d(Ax)) = tr(2x^TA^TAdx)$$

$$\underbrace{d(Ax)}_{=Adx}$$

$$= tr\big((2A^TAx)^T dx\big) \Rightarrow \nabla f = 2A^TAX$$

$$f(x) = \|Ax - b\|_2^2 + \gamma\|x\|_2^2$$

$$f_2 = \|Ax-b\|_2^2 = tr(2(Ax-b)^T d(Ax-b)) = tr(2(x^TA^T - b^T)Adx) = tr((2A^T(Ax-b))^T dx)$$

$$\Rightarrow \nabla f_2 = 2A^T(Ax-b)$$

$$\Rightarrow \nabla f = 2A^T(Ax-b) + 2\gamma X$$

گرادیان خطی است.

ب) الگوریتم کاهش گرادیان :

Given:

$x_0,\ \alpha > 0$ → step size

Initialize

$x \leftarrow x_0$

Repeat until convergence:

$$x \leftarrow x - \alpha(2x)$$

Given:

$x_0,\ \alpha > 0$ → step size

Initialize

$x \leftarrow x_0$

Repeat until convergence:

$$x \leftarrow x - 2\alpha A^T A x$$

Given:

$x_0,\ \alpha > 0$ → step size

Initialize

$x \leftarrow x_0$

Repeat until convergence:

$$x \leftarrow x - \alpha\left(2A^T(Ax-b) + 2\gamma x\right)$$

تعیین به گروه ها مختلف

یک حالت تعیین epoch

می تواند باشد که

for $t=1$ to T do

output x_t

$g_t = 2A^T(Ax-b) + 2\gamma x$

$x_{t+1} = x_t - \alpha g_t$

ج)

معادل ماتریس هسیان مثبت نیمه معین است :

بخش الف

$$\nabla g = 2x \longrightarrow \frac{\partial \nabla g}{\partial x_j} = (0, \dots, 2, 0, \dots)$$

$$\nabla(2x) = (2, \dots, 2)$$

$$\Rightarrow H_f = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & 0 & \cdots & 2 \end{pmatrix} = 2I_n$$

$$v^T 2I_n v = 2v^T v = 2\|v\|_2^2 > 0$$

$v \neq 0$

د) f محدب $\Rightarrow f(\theta x+(1-\theta)y) \leq \theta f(x)+(1-\theta)f(y)$ (*)

$$1=\theta+(1-\theta)$$

$$g(\theta x+(1-\theta)y) = f(A(\theta x+(1-\theta)y)-b) = f(\theta Ax+(1-\theta)Ay-b) =$$

طبق شرط محدب

$$f(\theta(Ax-b)+(1-\theta)(Ay-b)) \overset{*}{\leq} \theta f(Ax-b)+(1-\theta)f(Ay-b) = \theta g(x)+(1-\theta)g(y)$$

$\boxed{\checkmark} \leftarrow g$ محدب است.

ه)

$$g(x)=\|x\|_2^2$$

$$f(x) = \|Ax-b\|_2^2 = g(Ax-b) \quad (*)$$

طبق د) اگر g محدب باشد، $g(Ax-b)$ محدب است.

طبق ج) g محدب است.

$\Longleftarrow$ $g(Ax-b)$ محدب است.

$\Downarrow (*)$

$f(x)=g(Ax-b)$ محدب است.