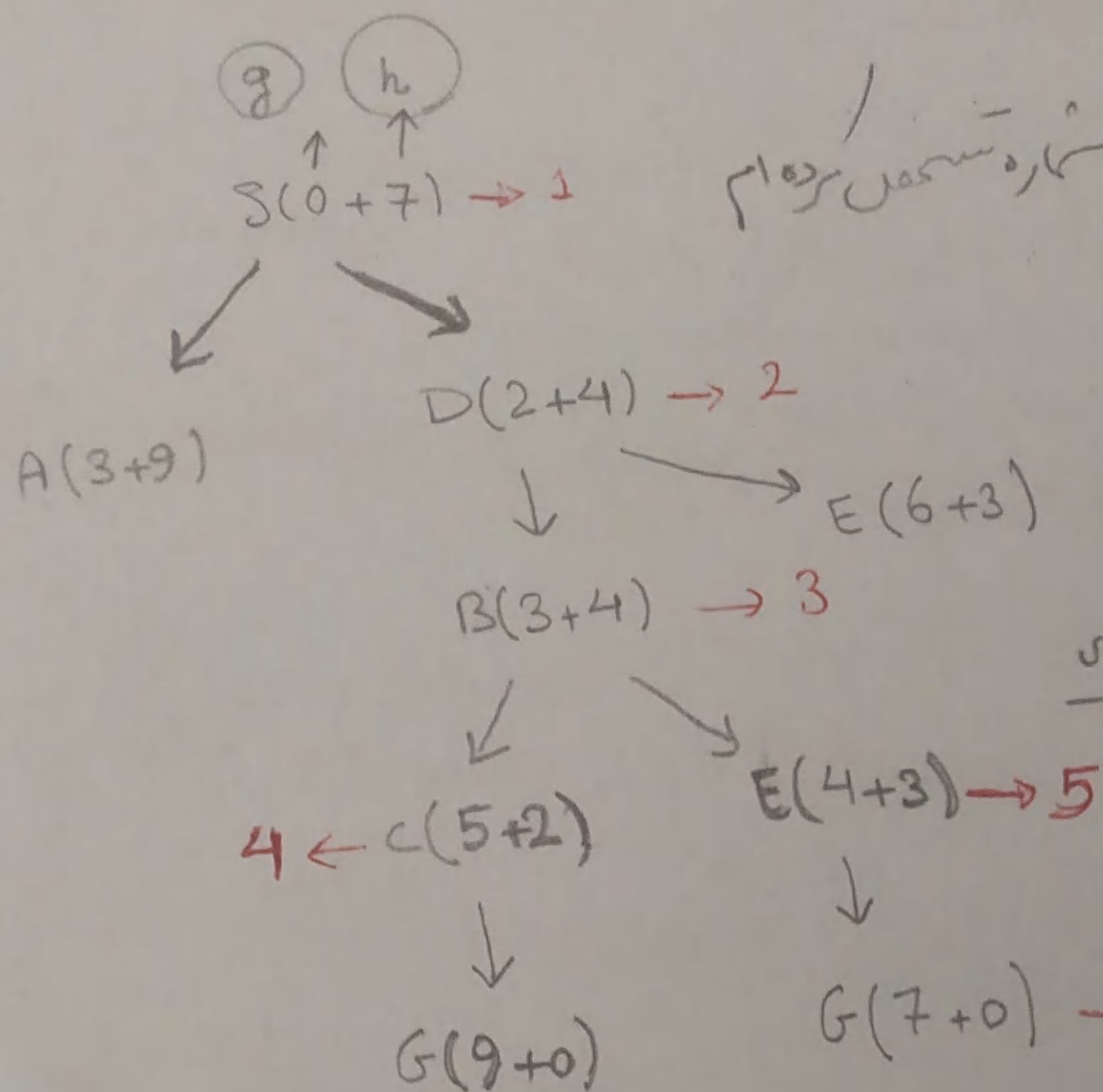


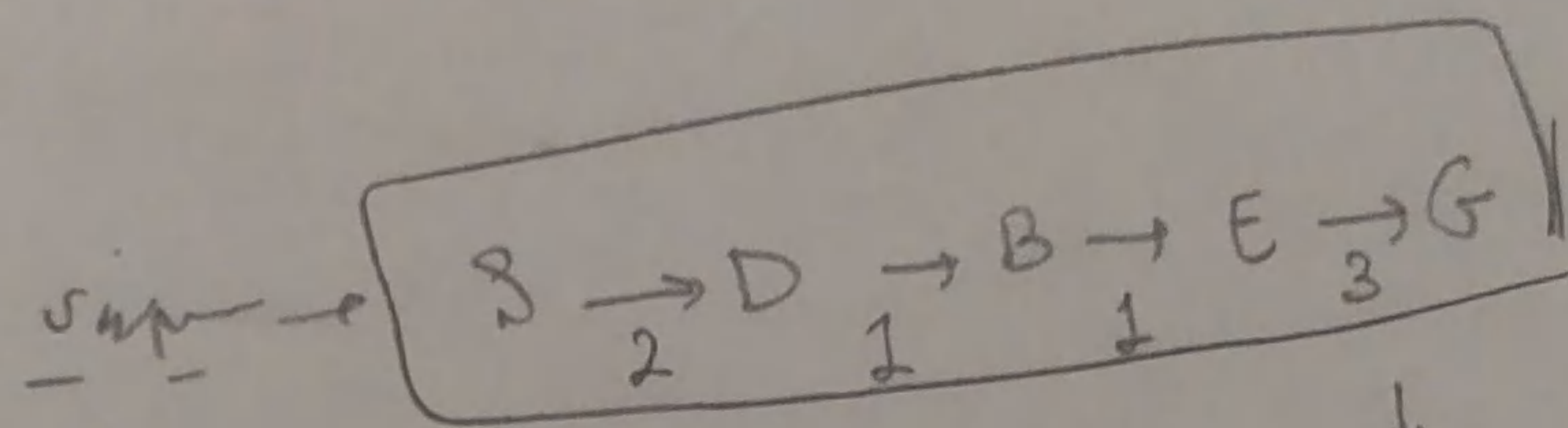
مقدار A* مقدار ①

$$f(n) = h(n) + g(n)$$



رابطه بین node و مقدار expand و مقدار h و مقدار g

مقدار h و مقدار g



مقدار h و مقدار g

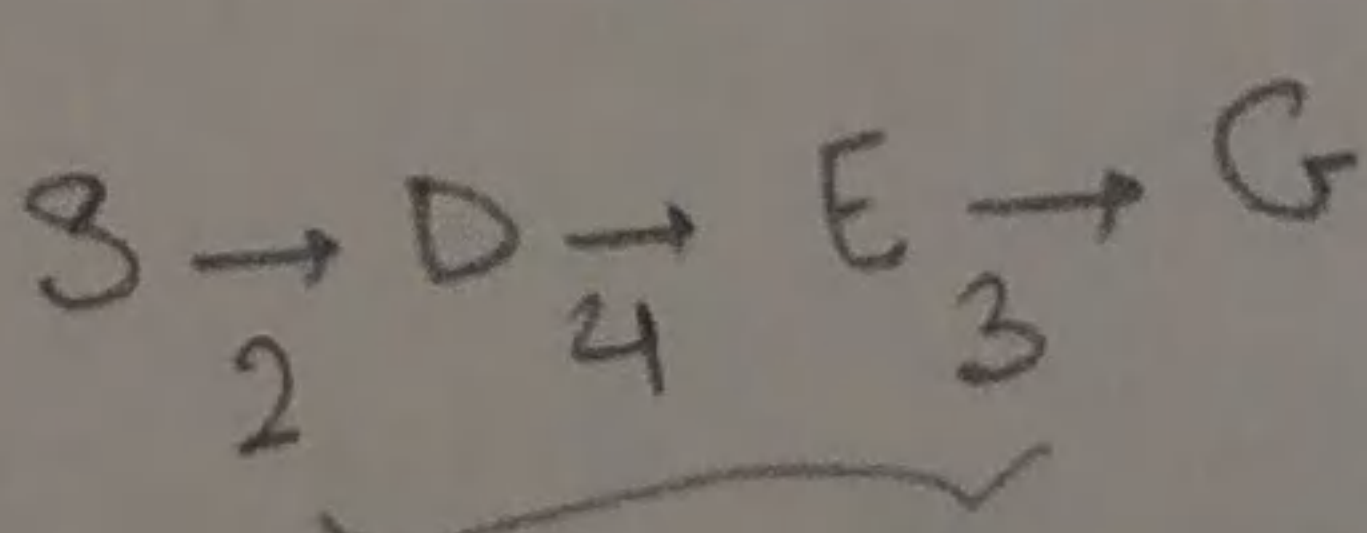
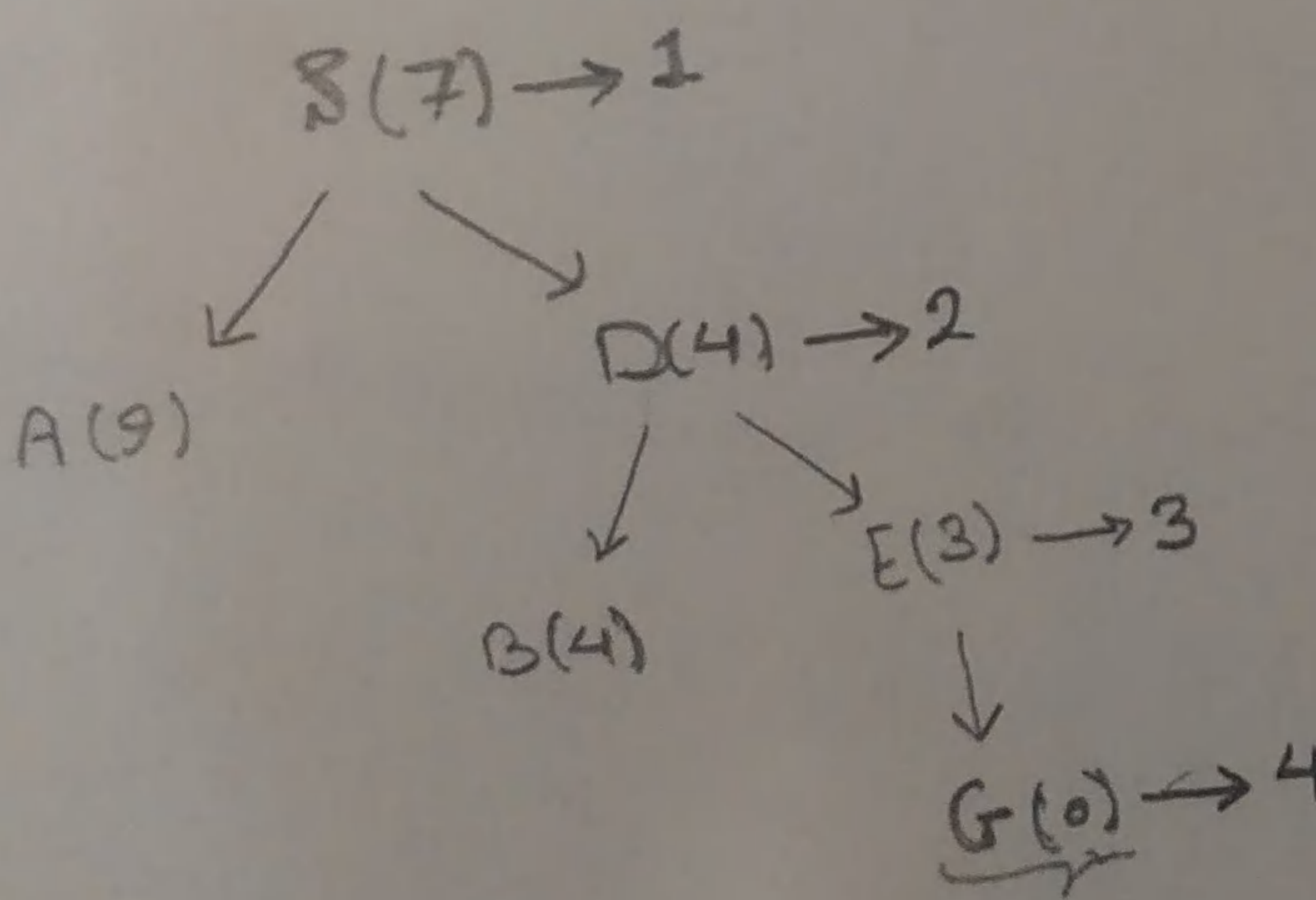
Goal

greedy

greedy

greedy

greedy



greedy

① 2-61

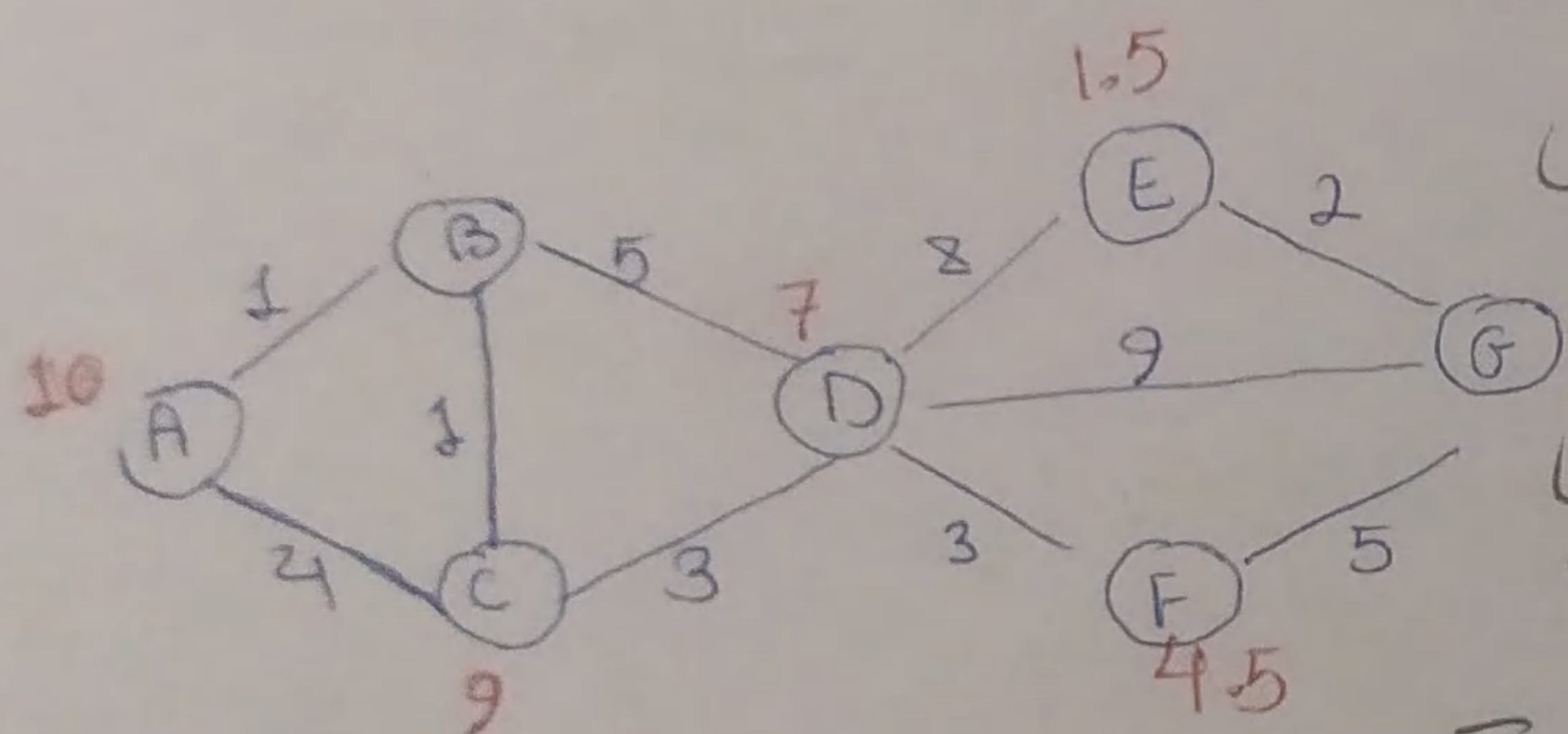
SPACE 24 Priority
queue

Time Complexity: $O(b^d)$

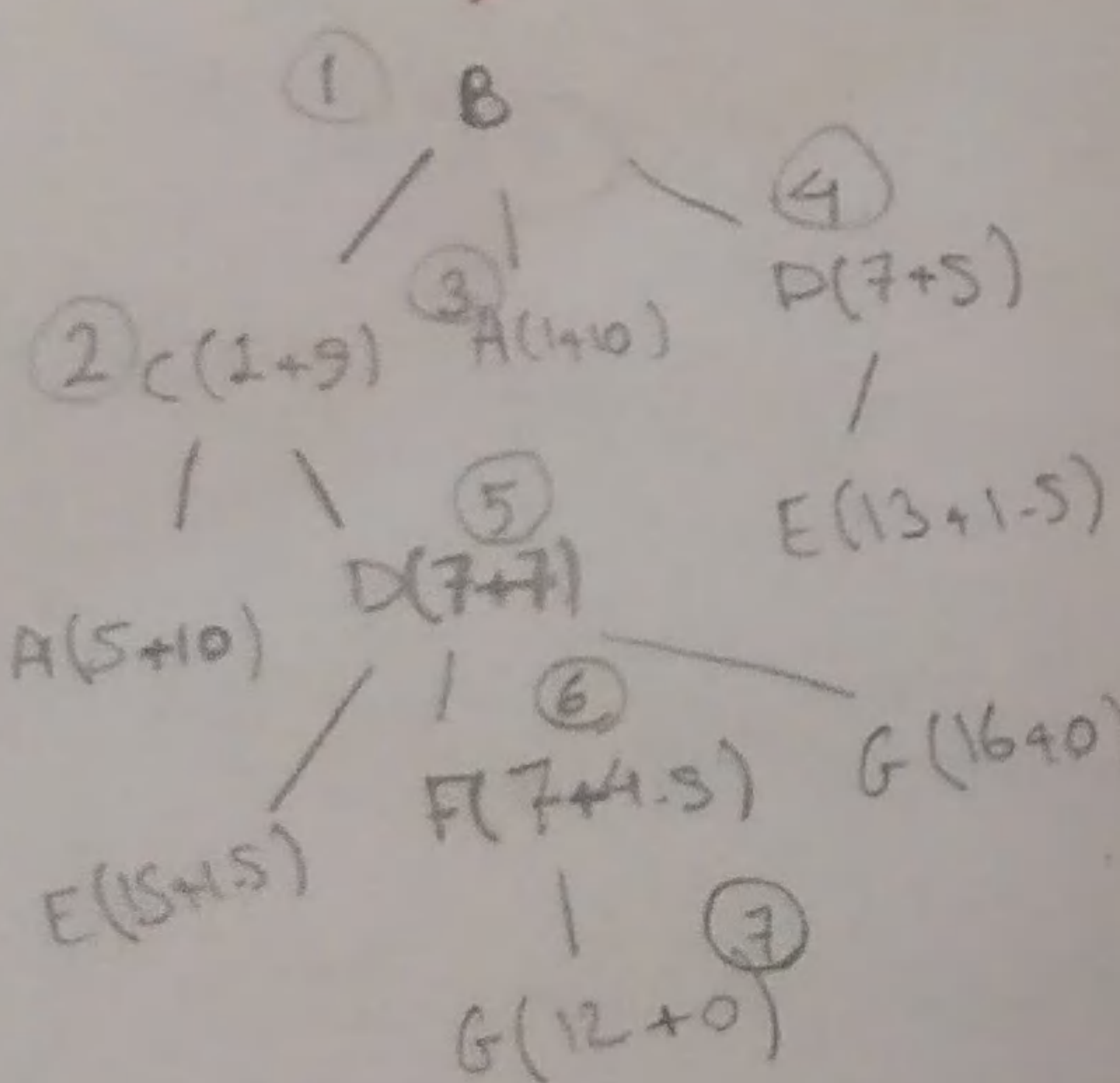
2) قال

Upper admissible σ_{max} and admissible $h_1(B)$ $\sigma_{\text{max}} \sigma_{\text{min}}$ ①

$h^*(B) = \sup_{A \in \mathcal{A}} h(A)$

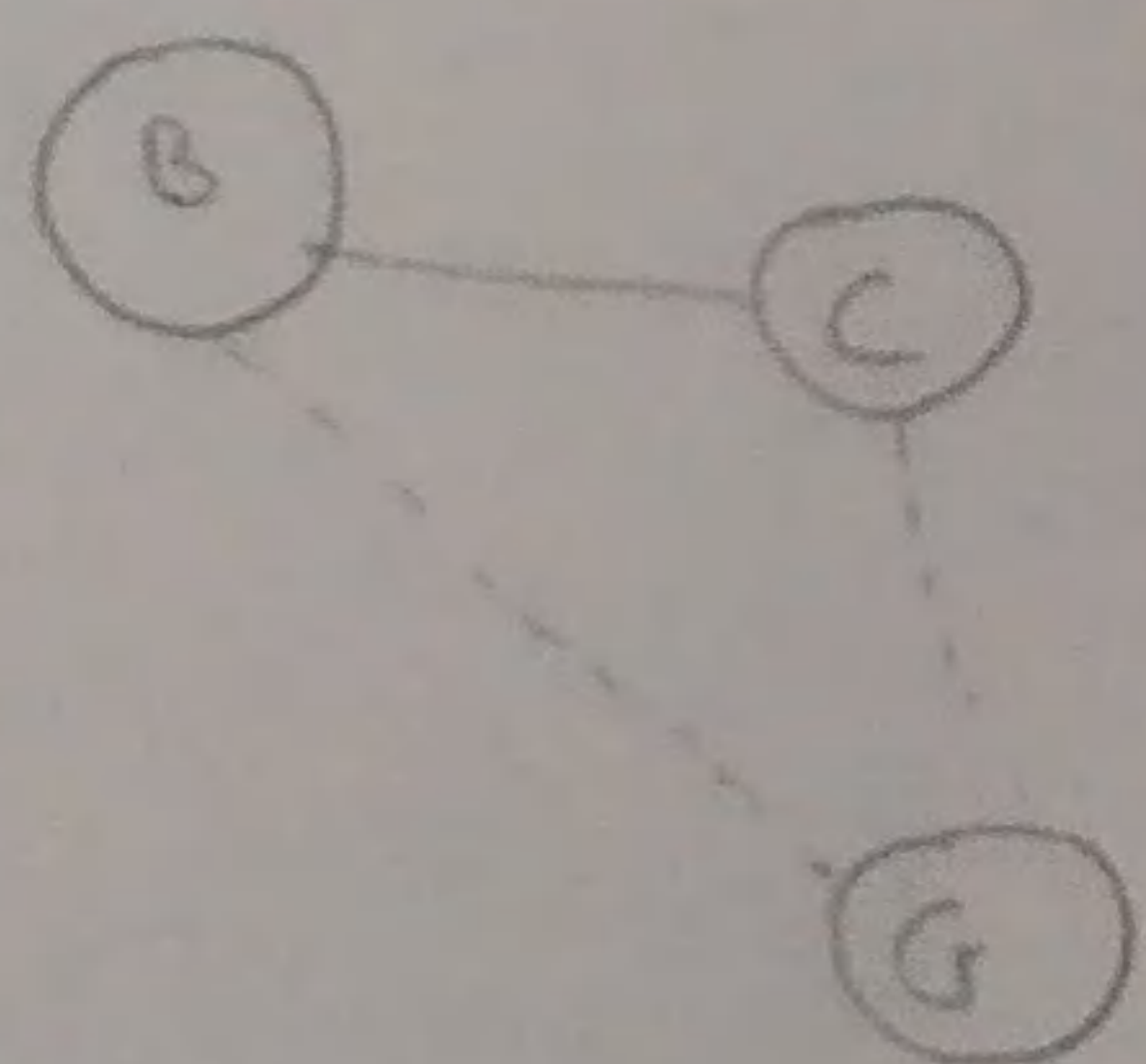


است G . B G برابر است 12 است



$$B \xrightarrow{1} C \xrightarrow{3} D \xrightarrow{3} E \xrightarrow{5} G \xrightarrow{\sqrt{10}} H$$

$\mathcal{L}_1 \boxed{0 \leq h_1(B) \leq 12} \cup \mathcal{L}_1$, admissible, h_1 μ_1 σ_1



در
A, B
مقدار

$$h(B) - h(A) \leq 1 \rightarrow h(B) \leq 10 + 1 \Rightarrow \boxed{9 \leq h(B) \leq 11} \quad (1)$$

$$h(A) - h(B) \leq 1 \rightarrow \frac{h(A)-1}{9} \leq h(B)$$

در
B, C
مقدار

$$h(B) - h(C) \leq 1 \rightarrow h(B) \leq 1 + 9 \Rightarrow \boxed{8 \leq h(B) \leq 10} \quad (2)$$

$$h(C) - h(B) \leq 1 \rightarrow \frac{h(C)-1}{8} \leq h(B)$$

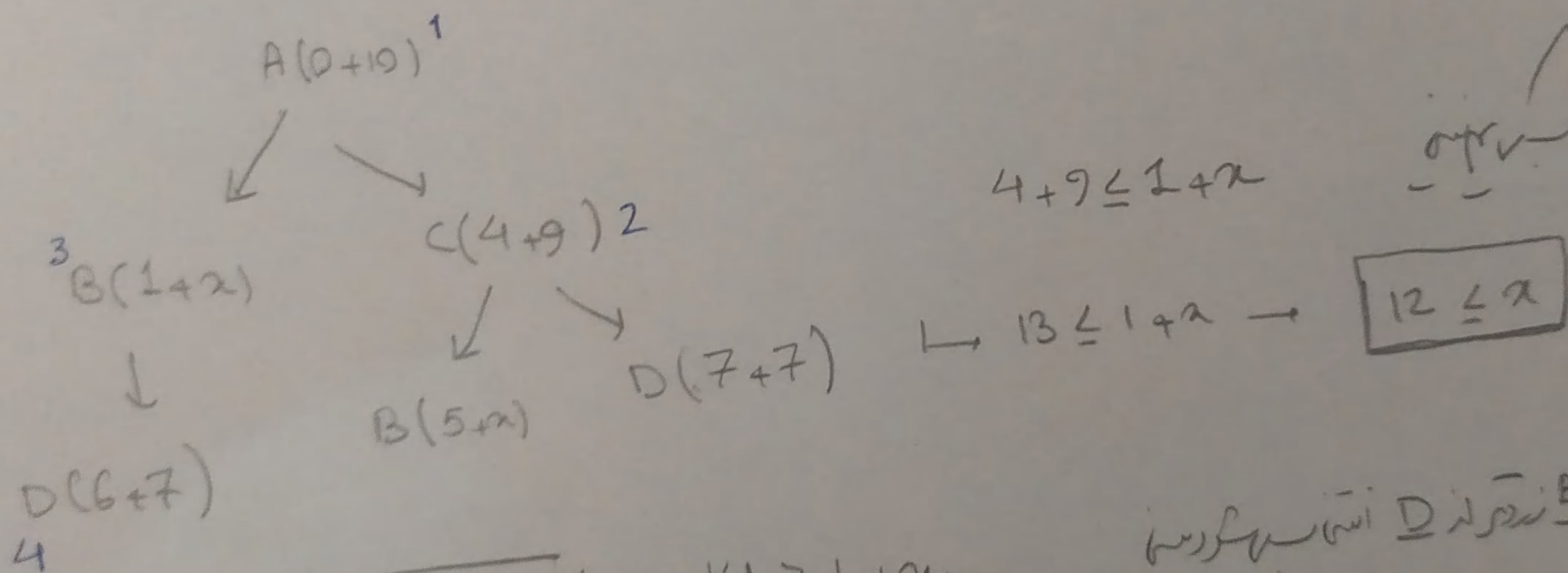
در
B, D
مقدار

$$h(B) - h(D) \leq 5 \rightarrow h(B) \leq 12 \rightarrow \boxed{2 \leq h(B) \leq 12} \quad (3)$$

$$h(D) - h(B) \leq 5 \quad 2 \leq h(B)$$

از استنتاج (1), (2), (3) نتیجه میگیریم $\boxed{9 \leq h_1(B) \leq 10}$ Consistent

(4) برای چیدمان $A^* \rightarrow C \rightarrow B \rightarrow D$ باید $h(B) = x$ را در نظر بگیریم



این منجر به $\boxed{9 \leq h_2(B) \leq 14} \leftarrow 14 \geq 1+x$ می شود

همچنین باید B و D استقراری داشته باشند
 بنابراین $\boxed{12 \leq h_2(B) \leq 13}$ است

توضیح نهاده می شود

این کلاس backtrack در کلاس Sudoku-CPSP است. این کلاس متد getNextLocation

دارد که این متد به کار می رود تا بررسی کند که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد isSafe نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

این کلاس CSP نام دارد. این کلاس متد solve نام دارد که در آن یک سطر یا ستون یا یک 3x3 مربع را بررسی می کند.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

در این متد بررسی می شود که آیا یک خانه در یک سطر یا ستون یا در یک 3x3 مربع امن است یا نه. این متد Valid نام دارد.

این کلاس CSP نام دارد. این کلاس متد solve نام دارد که در آن یک سطر یا ستون یا یک 3x3 مربع را بررسی می کند.

① → توانستیم از Arc Consistency استفاده کنیم و این است Forward checking

مقادیر مجاز یک خانه را می بینیم و می بینیم که در این خانه مقادیر مجاز را داریم یا نه

می بینیم که در این خانه مقادیر مجاز را داریم یا نه و می بینیم که در این خانه مقادیر مجاز را داریم یا نه

می بینیم که در این خانه مقادیر مجاز را داریم یا نه و می بینیم که در این خانه مقادیر مجاز را داریم یا نه

ما خانه ها را جدا جدا حساب می کنیم، رابطه بین خانه ها را می بینیم. در Arc Consistency، این کار را نمی توانیم انجام دهیم.

② → توانستیم از ordering استفاده کنیم و این کار را می توانیم انجام دهیم.

این کار را می توانیم انجام دهیم و می بینیم که در این خانه مقادیر مجاز را داریم یا نه

در این کار، ما می بینیم که در این خانه مقادیر مجاز را داریم یا نه و می بینیم که در این خانه مقادیر مجاز را داریم یا نه

این کار را می توانیم انجام دهیم و می بینیم که در این خانه مقادیر مجاز را داریم یا نه

Least Constraining Value استفاده می کنیم