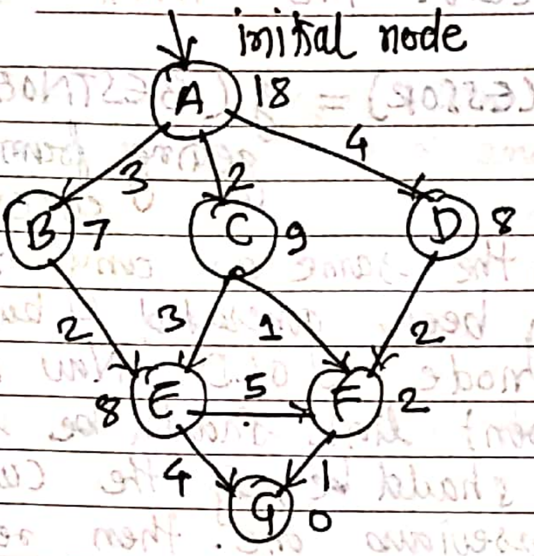
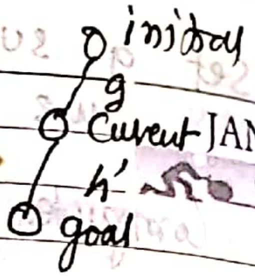
# Trace for A*





h- is written outside the node.

g-

(underestimated)

**Step.1**. OPEN → $\boxed{A}$ 18 CLOSED → empty list.

**Step.2** Best node is A since it is anly the node on OPEN. Remove it from OPEN, place it on CLOSED, generate successors of best node.

Compute, $f'(D) = 4+8=12$

$f'(B) = 3+7=10$, $f'(C) = 2+9=11$

∴ Since B, C & D are not either on OPEN or CLOSED thus add them on OPEN.

$$\text{OPEN} \rightarrow \overset{A}{\underset{10}{B}} \; \overset{A}{\underset{11}{C}} \; \overset{A}{\underset{12}{D}} \qquad \text{CLOSED} \rightarrow \underset{18}{\boxed{A}}$$

**Step3**. Best node is B. Now successors of B are A & E.

$\therefore f(A) = g(B) + h'(A).$

$= 3+3+18 = 24.$

Since successor A is already on CLOSED. Thus call it CLOSED OLD. Since new path is not better than OLD path, we will not change parent, g & f' value of A.

second successor is 'E' which is neither present on OPEN nor on CLOSED, thus add it to OPEN & Compute $f(E) = 3+2+8 = 13$.

OPEN →

| | | |
|---|---|---|
| A | A | B |
| C | D | E |
| 11 | 12 | 13 |

CLOSED →

| | |
|---|---|
| - | A |
| A | B |
| 18 | 10 |

**Step. 4.** Best node is C. Remove it from OPEN. & add it to CLOSED. Generate successors of C, which are A, E & F.

Since A is already present on CLOSED, thus we Call it as <u>CLOSED OLD</u>. & Compute $f'(A) = 2+18+2$

$f'(A) = 22$.

thus new path is not better than old one. So do not change parent & $f'$ of A.

As E is $f'(E) = 2+3+8 = 13$ & already present on OPEN, Call it OLD. Since both OLD & new paths are of equal cost. we will not change parent, $f'$ & link.

Now, Node F, which is neither present on OPEN or CLOSED, thus add it on OPEN & Compute $f(F) = 2+1+2 = 5$.

OPEN →

| | | |
|---|---|---|
| A | B | C |
| D | E | F |
| 12 | 13 | 5 |

CLOSED →

| | | |
|---|---|---|
| - | A | A |
| A | B | C |
| 18 | 10 | 11 |

**Step. 5** Best node is F. Remove it from OPEN, add it on CLOSED. Generate its successors which are C, D & G.

As C is already present on CLOSED, Call it as <u>CLOSED OLD</u>. $f(C) = 2+1+1+9 = 13$, which is worse than

previous path so do not change, parent, link & $f'$.

Node D is already present an OPEN, Call it OLD

$f'(D) = 2+1+2+8 = 13$; thus new path is not better than old one, so do not change anything like parent, f' & link.

As G is a successor, which is neither on OPEN nor on CLOSED, thus add it to OPEN & Compute

$f'(G) = 4+0 = 4.$

| | | | |
|---|---|---|---|
| | A | B | E |
| OPEN → | D | E | G |
| | 12 | 13 | 4 |

| | | | | |
|---|---|---|---|---|
| CLOSED → | A | B | C | F |
| | 18 | 10 | 11 | 5. |

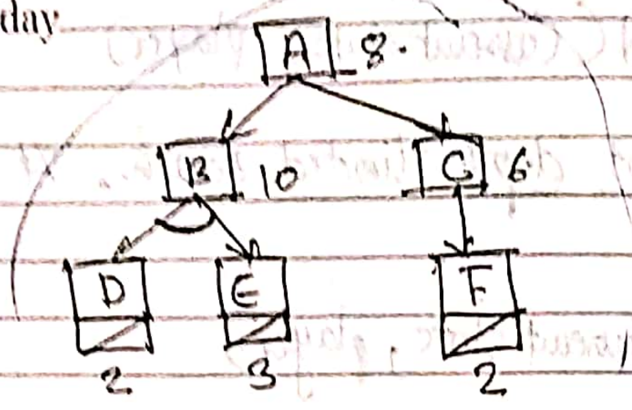
(Step 6.) Best node is G. Since it is goal state as (h'=0). algorithm has to halt by reporting success with a goal node G & Cost=G & path.

= G, F, C, A

Web references:- Learn about A*

1. http://en.wikipedia.org/wiki/A-star_Search_algorithm.

# New (Final) Trace of AO*



initialize:- FUTILITY = (+∞)

— terminal nodes

1- cost of each arc.

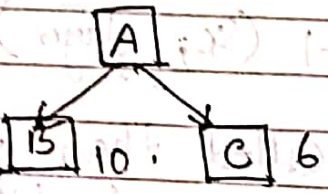
step 1. GRAPH INIT = A

[A] h'(init) = 8.

step 2. (a) since there is only node A on ~~goo~~ GRAPH. Since A is unexpanded node, select it for expansion & call this as NODE

∴ NODE = A.

(b) Generate successors of NODE = A. i.e B & C



add them on GRAPH if they are not also ancestors of NODE.

(c) (i). initialize S = {A}

since there is only node ~~A~~ A in S

call it CURRENT = A.

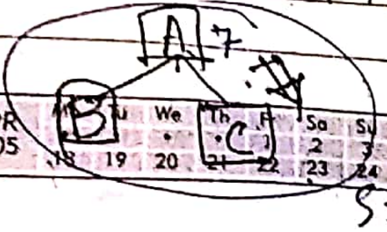
(ii) Compute cost of each arc emerging from CURRENT.

Cost(AB) = 10+1 = 11.

Cost(AC) = 6+1 = 7.

change h'(CURRENT) = (7) = h'(A)

(iii) Mark best path out of CURRENT.



S = {B, C}

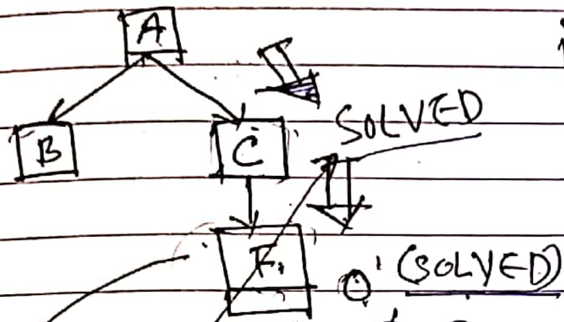
Step 2 (a). Trace the labeled arc (best Path) from INIT & select for expansion an unexpanded node. Now the node is C. Call it NODE = C.

(b) Generate successors of NODE = C.

i. Add successors to GRAPH.

ii. SUCCESSOR is terminal node

∴ $h'(F) = 0$



(c) S = { F, ~~C, A~~ }

(i) CURRENT = F

(iv) since CURRENT is labelled as solved add ancestors of F to GRAPH.

Saturday 23
113 - 252

S = { C, A }

2(c) (i) CURRENT = C

$h'(\text{CURRENT}) = h'(C) = 0+1 = 1.$

(iv) Mark C as SOLVED

2(c) (i) S = { A }

CURRENT = A.

Sunday 24

(ii) $h'(\text{CURRENT}) = h'(A) = 1+1 = 2$

(iv) Mark A "SOLVED" since all its childs are marked SOLVED.

SINCE INIT is labeled SOLVED algorithm will halt.