Heuristic Search

Best First Search:

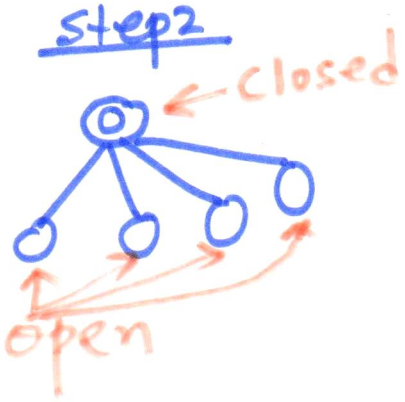
* heuristic function $h(n)$
  — estimate distance to the goal
* It sorts the open list.

$$OPEN \leftarrow Sort_n(append\ New\ tail\ (OPEN))$$

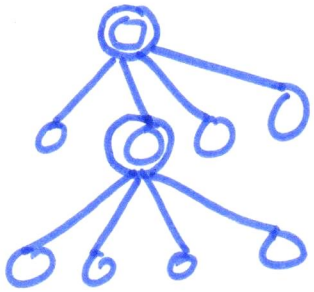
[PRIORITY QUEUE]

step1

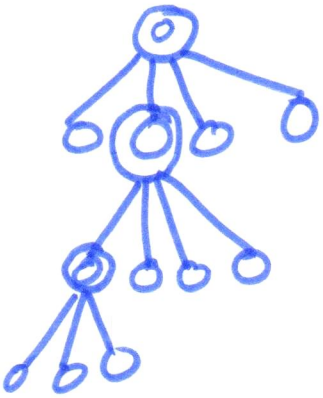
step2



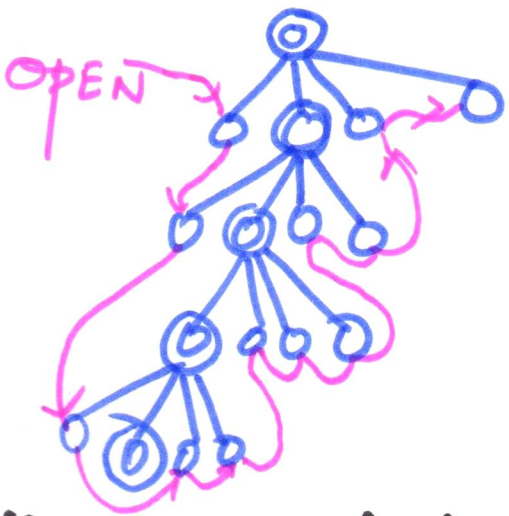
step3



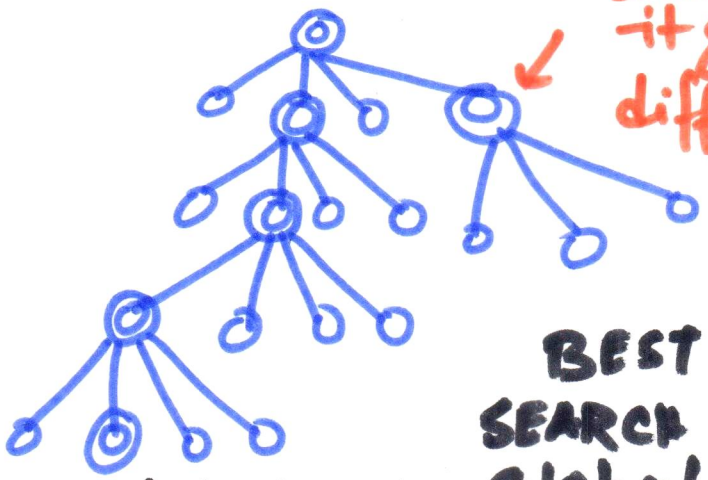
step4



step5



step5



Entire OPEN List is global List.

BEST FIRST SEARCH IS global search algorithm

Time: } depends upon $h(n)$ to be exponential.
Space: }

Quality: It is complete the quality of the solution as we said, it is not necessarily an optimal path.

Completness: It's complete (for finite space)

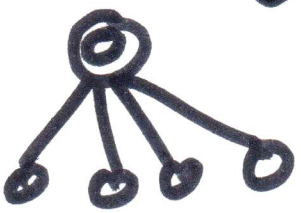
→ Not guarantee

Hill climbing (Local search)

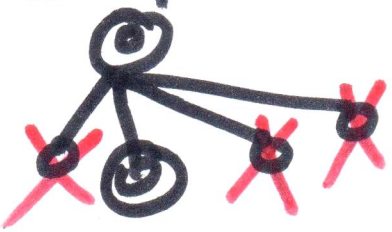
OPEN ← $Sort_n$ (New)

step1

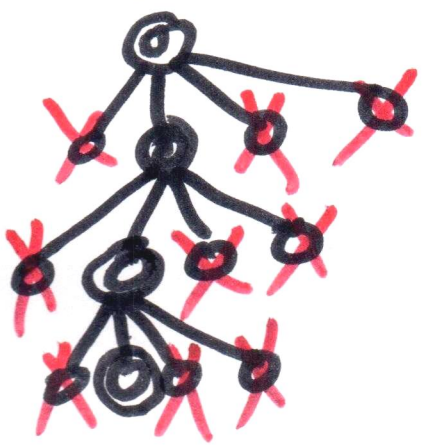
step2



step3



step4



```
while Next is better than Current
{
  Next ← Best(Movegen(corent))
}
```

Termination Criteria:

Best first Search: (Global Search)

* goal test
* OPEN Empty

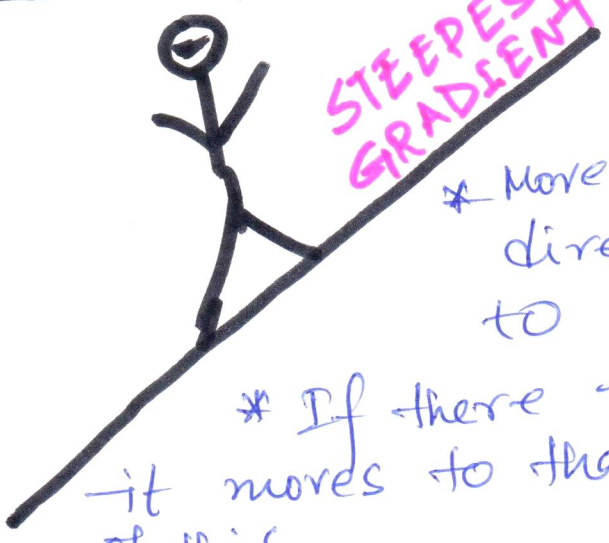
} terminate.

Hill Climbing (Local Search)

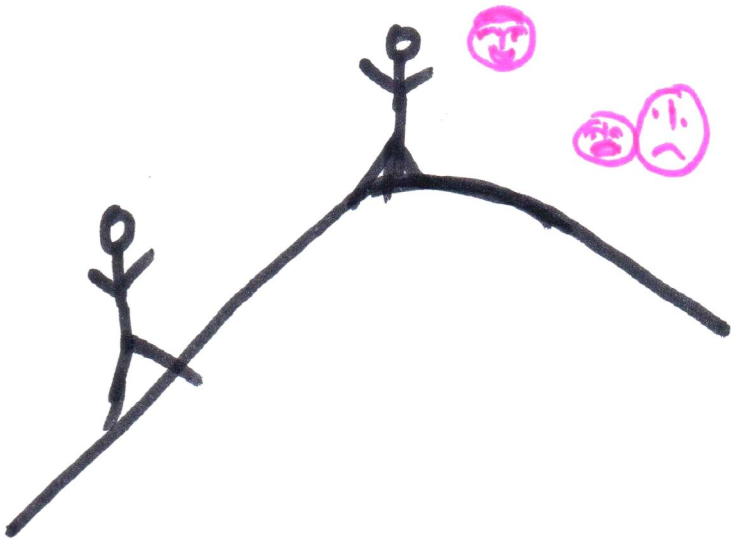
* Next is better than Current

– OPTIMIZATION – optimize the value of $h(x)$

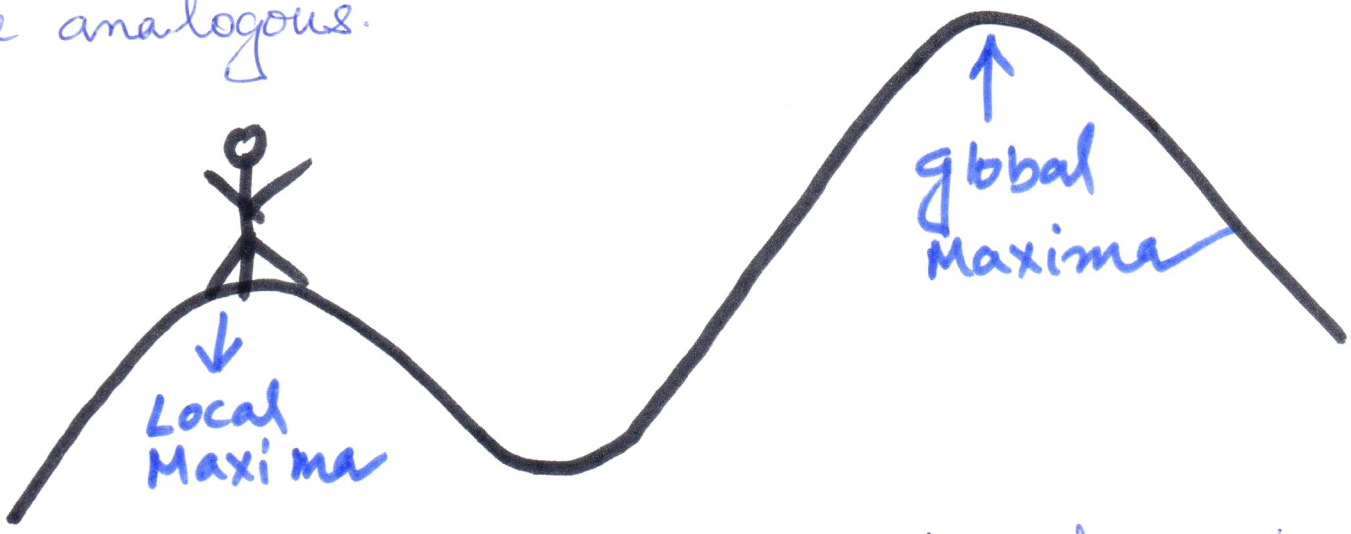
Hill Climbing



* Move forward to that direction which seems to be going up.
* If there is a better neighbor it moves to that state next is best of this.



* This algorithm will take it to maxima or a minima if you are minimizing because it should be analogous.



* you will get stuck in this Local maxima and this is the problem because of the fact that this algorithm is a <u>local search algorithm</u>. ← [it's looks only neighbourhood of the current state and decide]

When Hill climbing Algorithm will work best?

if the nature of the problem is such that the heuristic function defines a smooth and monotonic surface the hill climbing will work. Otherwise it will get stuck to a local Maxima.