

Search

Summary

- ▶ Search can be **informed** or **uninformed**
 - ▶ BFS, DFS, UCS are **uninformed**
 - ▶ A* and GBFS are **informed** (they use a heuristic function: a *guess*)
- ▶ Be mindful of the use of **memory** and the **implementation details**.
 - ▶ Do we remember places we have visited before?
 - ▶ Which function are we using to **prioritise** a node?
- ▶ Study when the **goal-check** is performed on different algorithms!
- ▶ Learn the difference between **admissible** and **consistent** heuristics and what they imply
 - ▶ **Admissible**: $h(n) \leq h^*(n)$ for all n
 - ▶ **Consistent**: $h(n) \leq c(n, n') + h(n')$ for all n and its children n'
- ▶ No need to study iterated deepening search

Search

Example question

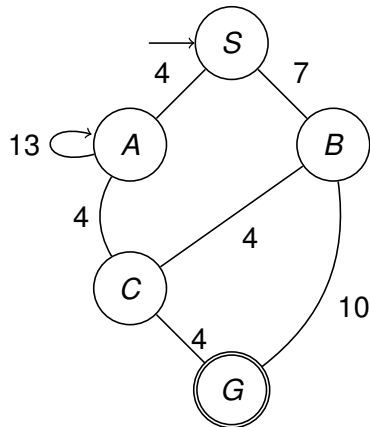
What is the order of nodes expanded by *<ALGORITHM>* on the following graph?

With the following estimated distances to the goal:

- ▶ $h(A) = 3$
- ▶ $h(B) = 3$
- ▶ $h(C) = 3$
- ▶ $h(G) = 0$

What happens if $h(B) = x$?

What about $h(B) \geq y$?



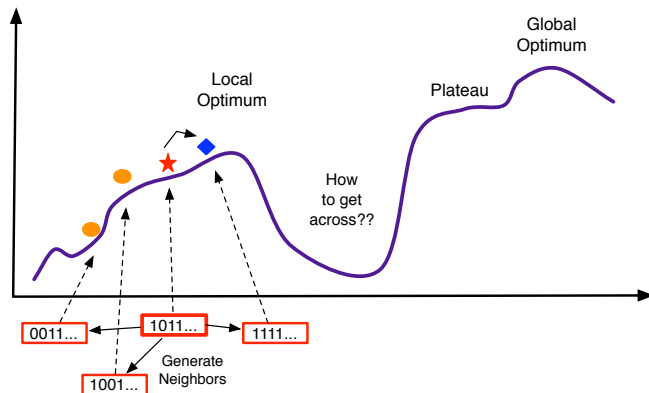
Remember to consider that nodes could be expanded more than once!

Local Search

Summary

Sometimes we only need to find a **good enough** solution, so we can search **locally**: go to the best spot you see now.

- ▶ Assume you are doing **maximisation**
- ▶ You then want to **climb** the tallest peak
- ▶ This is called **hill-climbing**!



If you are **minimising** instead, then the procedure is called **gradient descent** as we want to move towards the direction where the difference in “height” is largest.

Complex Environments (and a bit of planning)

Summary

- ▶ Learn **how to get across** valleys and plateaus in the **search space landscape**
- ▶ Consider what happens to our search (or our plan) when exploring **complex environments**
 - ▶ I cannot see (sensorless)
 - ▶ I can see, but I am not sure what will happen (contingency/uncertainty)
 - ▶ I need to keep an eye (online/real-time)
- ▶ Remember that states could become **belief states** instead!
- ▶ Instead of a single action between states, we may end up with a **function** that gives us the probability of reaching different states