AI 2002 Artificial Intelligence

Heuristic Search Algorithms I
(Informed Search)

Lecture 4

Department of Computer Science
National University of Computers & Emerging Sciences
Lahore.

Things Learnt So Far …

- What are States and State Spaces?
- Problems and their State Space Representation
- Graph Theoretical Concepts
- Strategies for State Space Search
  - Data Driven Search
  - Goal Driven Search
- Search Methods
  - Blind Search Methods
    - Depth First Search
    - Breadth First Search
    - Iterative Deepening Search
- Direction of Search

Measuring problem-solving performance

- **Completeness:** Is the algorithm guaranteed to find a solution when there is one?
- **Optimality:** Does the strategy find the optimal solution (lowest cost)?
- **Time complexity:** How long does it take to find a solution?
- **Space complexity:** How much memory is needed to perform the search?

Time and space complexity are always considered with respect to some measure of the problem difficulty. (State Space graph, branching factor, depth of the shallowest node, and number of nodes generated)

Comparison of Uniformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes^a	Yes^{a,b}	No	No	Yes^a	Yes^{a,d}
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes^c	Yes	No	No	Yes^c	Yes^{c,d}

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: ^{a} complete if b is finite; ^{b} complete if step costs $\geq \epsilon$ for positive ϵ; ^{c} optimal if step costs are all identical; ^{d} if both directions use breadth-first search.

In Today's Lecture …

- We will continue our discussion of Search Methods.
- We will learn what Heuristic Methods are and how can we apply them.
- Mainly we will study various commonly used heuristics in searching problems.

Today's Topics Are …

- General definitions and explanation of heuristic search methods (~ 5min).
- Hill Climbing
- Best-First Algorithm
- A* Search Algorithm
- A* Search with Simple Pruning
- Local Maxima, Ridges and Plateaus

Heuristic Search Methods

- Blind search techniques are so called because they use no information that may be available about the structure of the tree or availability of goals, or any other information that may help optimize the search. They simply trudge through the search space until a solution is found.
- A heuristic search makes use of available information in making the search more efficient. It uses heuristics, or rules-of-thumb to help decide which parts of a tree to examine. *A heuristic is a rule or method that almost always improves the decision process.*

Heuristic Search Methods

- For example, if in a shop with several checkouts, it is usually best to go to the one with the shortest queue.
- This holds true in most cases but further information could sway this - if you saw that there was a checkout with only one person in that queue but that the person currently at the checkout had three trolleys full of shopping and that at the fast-checkout all the people had only one item, you may choose to go to the fast-checkout instead. Also, you don't go to a checkout that doesn't have a cashier - it may have the shortest queue but you'll never get anywhere!!
- So heuristics are rules that usually help us but don't always give the best solution. They are fallible.
- A heuristic is only an informed guess of the next step to be taken in solving a problem. It is often based on experience or intuition.

When Heuristic Search Methods Are Used?

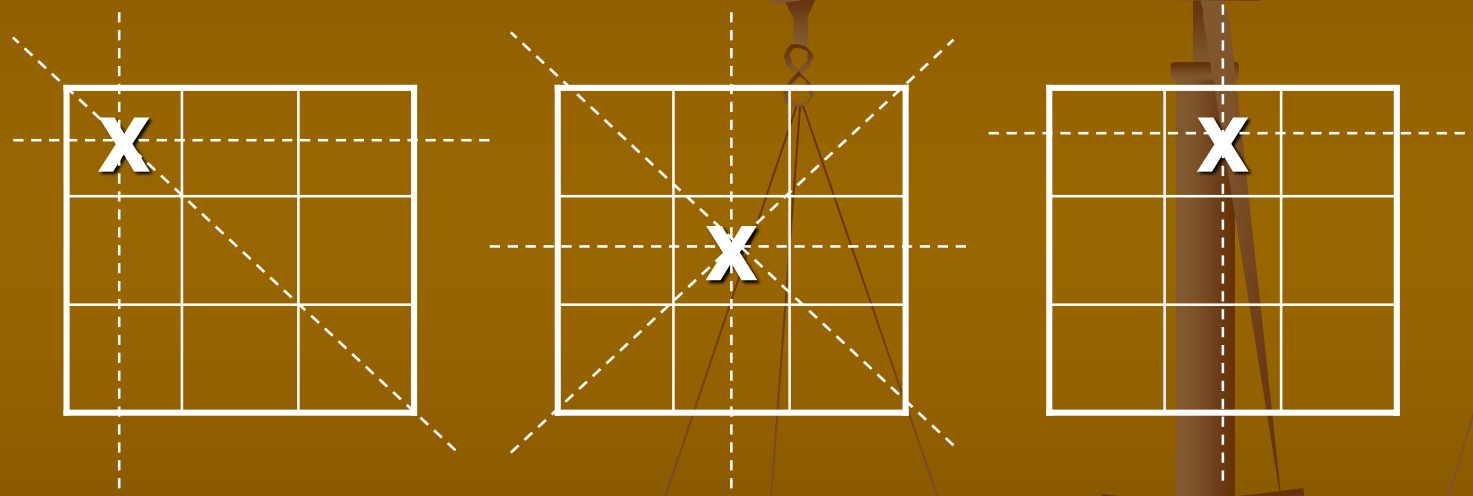
1. A Problem may not have an exact solution because of inherent ambiguities in the problem statement or available data.
   - Medical Diagnosis is an example of this.
     - A given set of symptoms may have several possible causes; Doctors choose the most likely diagnosis and a plan of treatment.
   - Vision is another inherently inexact problem.
     - Visual scenes are often ambiguous, allowing multiple interpretation of the connectedness, extent and orientation of objects. Optical illusions exemplify these ambiguities.
     - Vision systems use heuristics to select the most likely of several possible interpretations of a given scene.

When Heuristic Search Methods Are Used? (continued…)

2. A Problem may have an exact solution, but the computational cost of finding it may be prohibitive.
   - In many problems (e.g. chess), state space growth is explosive, with the number of possible states increasing exponentially or factorially with the depth of the search.
   - In such cases exhaustive (brute-force) search techniques such as depth-first or breadth-first may fail to find a solution within any practical length of time.
   - Heuristics attack this complexity by guiding the search along the most "promising" path through the space.
   - By eliminating states and their descendants from consideration, we hope to defeat this explosive growth of state space and find an acceptable answer.

How Heuristic Search Methods Can Be Used?

- A simple heuristic in the game of tic-tac-toe could be:
  We may move to the board in which X has the most winning lines



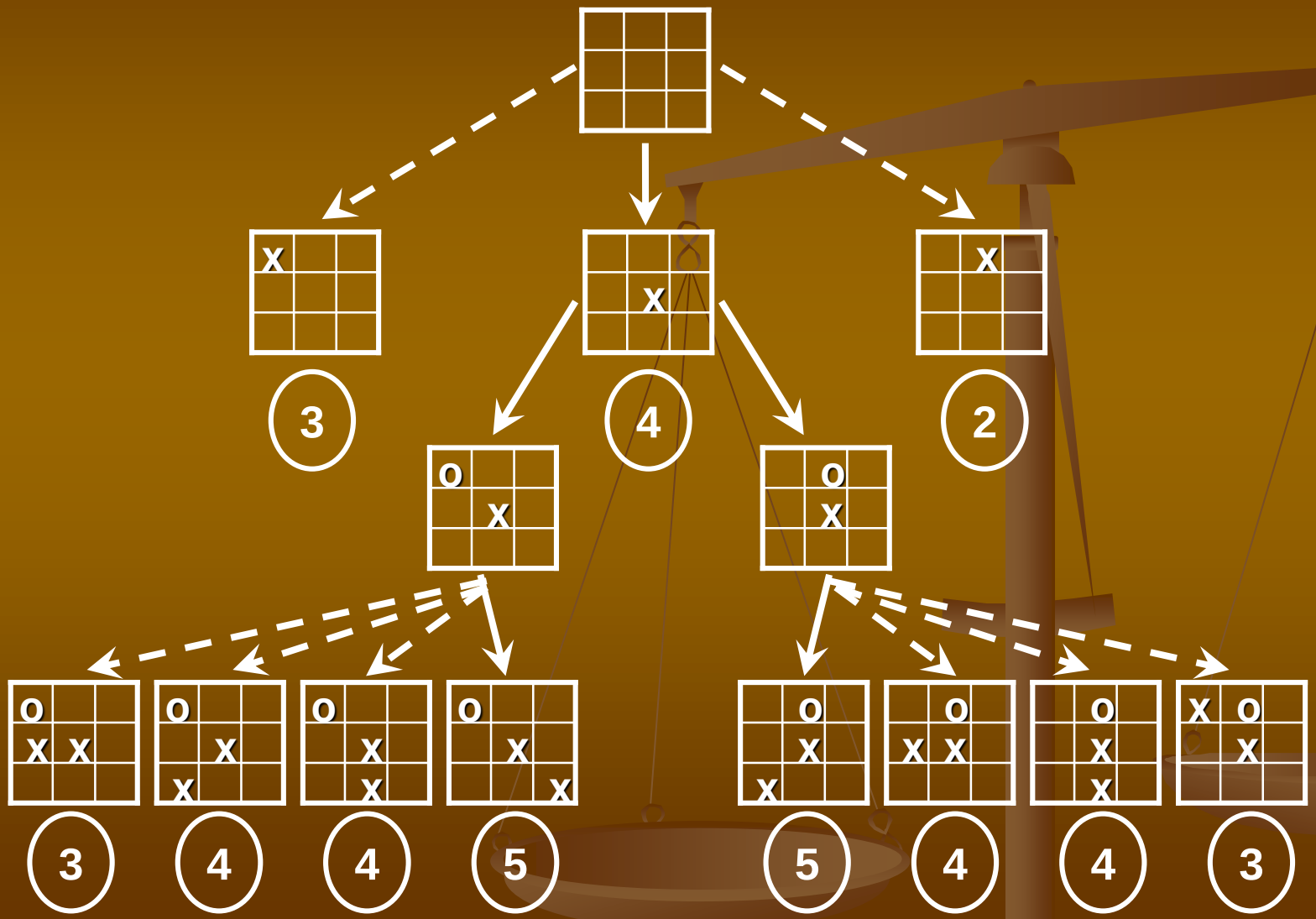
Three wins through a corner square

Four win through the center square

Two wins through a side square

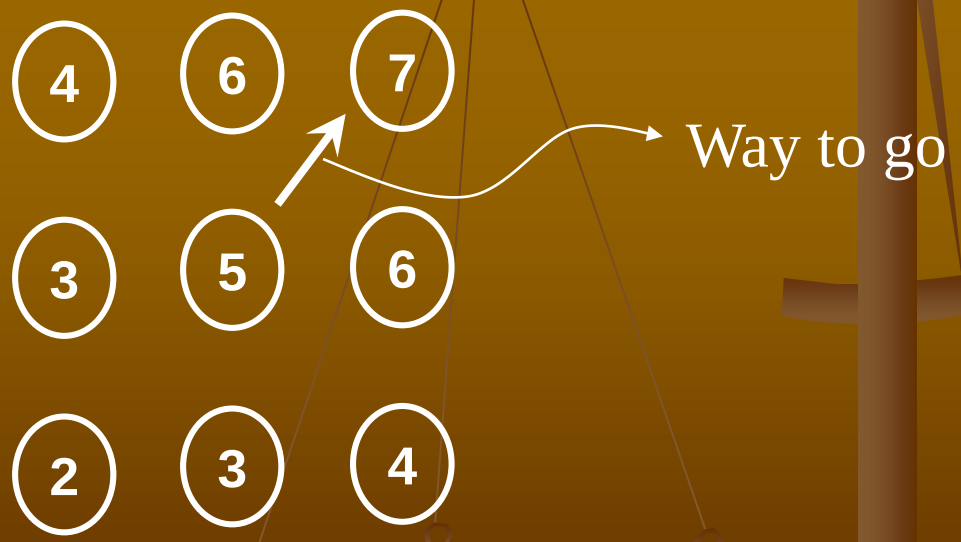
How Heuristic Search Methods Can Be Used?

- Heuristically reduced state space for tic-tac-toe



Hill Climbing Method

- In the case where only local knowledge is available - you have no idea how far away the goal is - the hill-climbing method may be a good solution.
- The method is best described as if you are trying to climb a hill in fog.
- You have an altimeter in your pocket and you want to go up.
- The way the hill-climbing heuristic works is to find the altitude of places near your current location and then going in the direction that gives you the highest upward gradient.
- The term local knowledge describes the fact that you know the altitude of places in your locality, the local terrain.



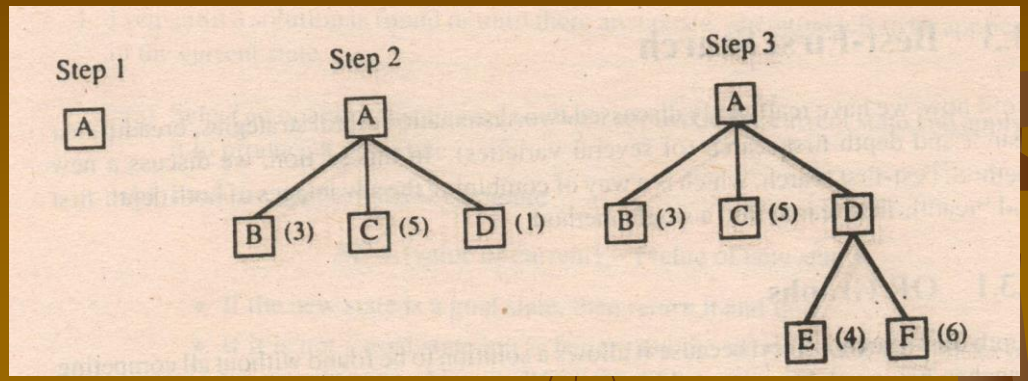
- The numbers in circles indicate the altitudes of neighboring places.

Best-First Algorithm

- The best first algorithm is a variation on the depth-first algorithm that uses heuristics.
- For each node we calculate some evaluation function to determine how good each node is.
- We order the stack so that the best nodes come off first and the worse nodes are left to be considered only if the more promising nodes fail to produce a solution.

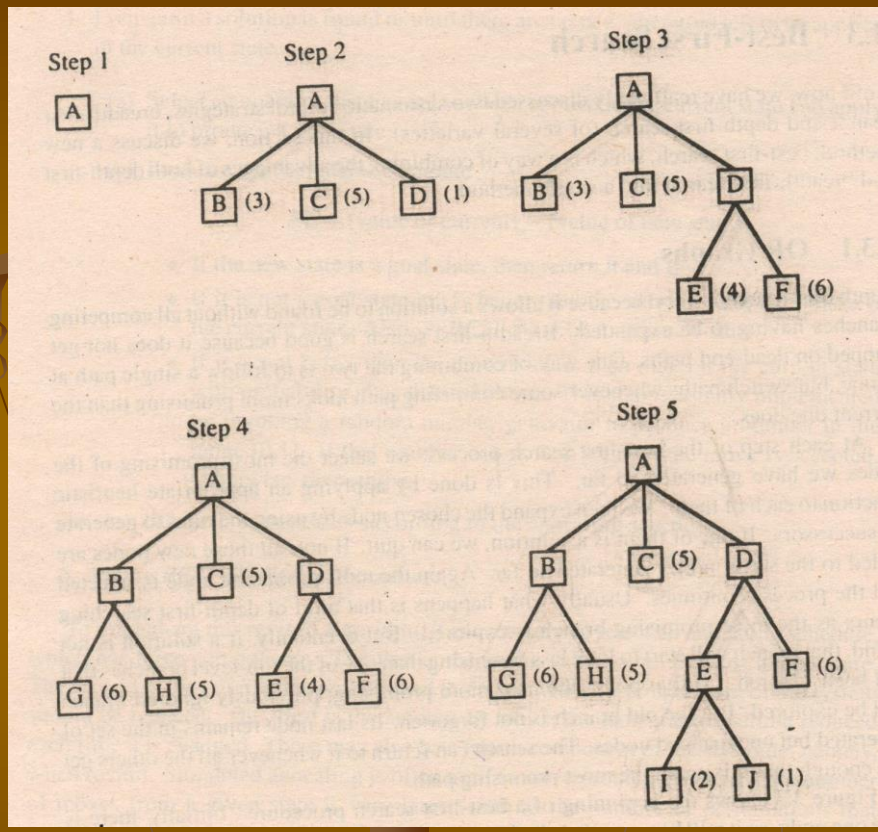
Best-First Search - Algorithm

1. Set N to be a sorted list of initial nodes.
2. If N is empty exit with failure.
3. Set n to be the first node in N and remove n from N.
4. If n is a goal node exit with success.
5. Otherwise, add the children of n to N.
6. Sort the nodes in N according to their estimated distance from a goal node.
7. Goto 2


Step 1
A
Step 2
A
B (3)
C (5)
D (1)
Step 3
A
B (3)
C (5)
D
E (4)
F (6)

Best-First Search - Example

- Initially there is only one node, so it will be expanded.
- Doing so generates three more nodes.
- The heuristic function in this example , is an estimate of the cost of getting to a solution from a given node.
- This heuristic is applied to each node.
- Since D is the most promising, it expanded next, producing two new successor nodes, E and F. The heuristic function is applied to them.
- Now another path, that going through node B, looks more promising, so it is pursued, generating nodes G and H.
- But again when theses nodes are evaluated, they look less promising.
- So attention is returned to the path through D to E.



Best-First vs. Hill Climbing

- This procedure looks very similar to hill climbing, with two exceptions:

1. In hill climbing, one move is selected and all other moves are rejected, never to be considered.

   - ***In best-first search, one move is selected, but the others are kept around so that they can be revisited later if the selected path becomes less promising.***

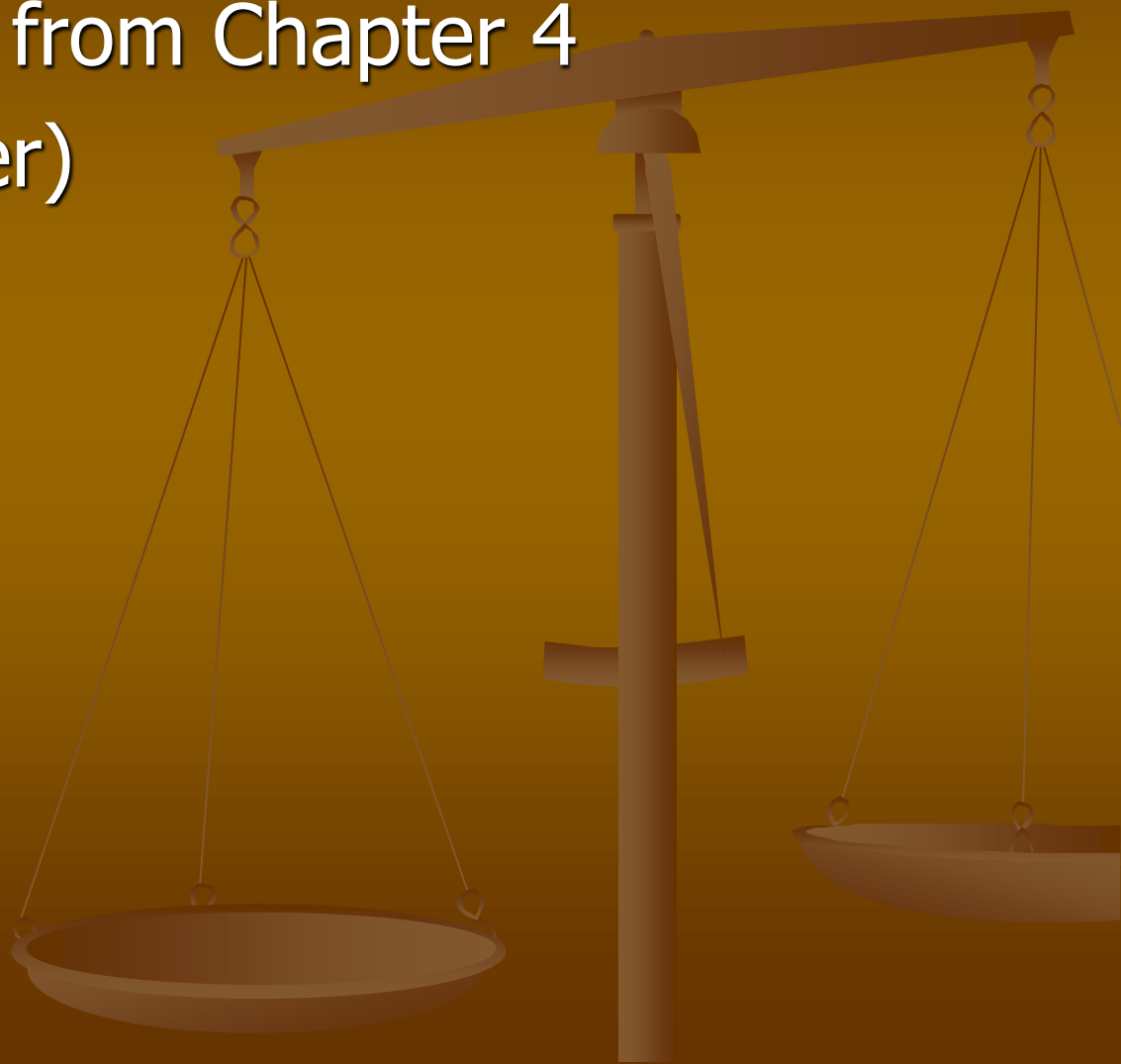
2. The best available state is selected in best-first search, even if that state has a value that is lower than the value of the state that was just explored.

   - ***This is in contrast with hill climbing, which will stop if there are no successor states with better values than the current state.***

Problems with the Hill Climbing Heuristic

This heuristic works well in many simple problems but can come into trouble in certain circumstances:-

- The Local Maximum
  - It is easy to conceive the problem where there are 'foot-hills' around the main peak which are local maxima. Once you get onto the top of one of these, all ways are down - there is no 'good' way to go, as far as the search algorithm knows, it has reached the top.
  - A simple technique to attempt to solve this problem is to perform a few searches and see if they end up at the same point - if they do you've probably found the goal. If they disagree then keeping the best one try doing some more searches and see if any of them produce better results. If they don't then it is reasonable, though not always correct, to assume that you have found the goal point.
- The Ridge
  - Assuming that we always take steps in either a north-south or east-west direction which is a reasonable assumption, then if we find ourselves on a ridge that runs north-east to south-west, or north-west to south-east, we will find that our four different direction steps all lead down! This is most severe in the cases where the sides of the ridge are steep and the climb of the ridge is slow.
  - Taking smaller steps or looking at diagonal steps may help solve the problem, but this is not always possible - especially with discrete data.
- The Plateau
  - Going from all downhill gradients to no gradients at all. On a plain all the points are at the same height and there is no obvious benefit in any direction.
  - The simple solution to this is to take larger steps, hoping that you will find an upward edge to the plain.

Home Work

- Read the topics from Chapter 4
- (George F. Luger)