CS 491 CAP
Advanced Search & Simulation

Jingbo Shang
University of Illinois at Urbana-Champaign
Dec 1, 2017
Outline

◊ Bitmasks for Pruning
◊ Search Order
◊ Bidirectional Search
◊ A* Search
Outline

◊ Bitmasks for Pruning
◊ Search Order
◊ Bidirectional Search
◊ A* Search
N Queens

◊ Find the total number of layouts of $n$ queens on an $n \times n$ chessboard, such that any two queens will not attack each other.

◊ Bruteforce:
  - Every time, put a queen at $(x, y)$
  - Mark $(\ast, y)$ and $(x + \Delta, y + \Delta)$ and $(x + \Delta, y - \Delta)$ as attacked
    • This step is $O(n)$
  - Recursion
Bitmask

◊ Query the $d$-th digit of $x$
  ▪ $(x \gg d) \& 1$
◊ Set the $d$-th digit of $x$ as 1
  ▪ $x \mid (1 \ll d)$
◊ Set the $d$-th digit of $x$ as 0
  ▪ $x \& (\neg(1 \ll d))$
◊ Only keep the last non-zero digit of $x$
  ▪ $x \& (\neg x)$
N Queens

◊ Mark columns, diagonals only
◊ It becomes $O(1)$

◊ Find the possible position in $x$-th row
◊ $\Leftrightarrow$ Find the non-zero bits
Outline

◊ Bitmasks for Pruning
◊ **Search Order**
◊ Bidirectional Search
◊ A* Search
Sudoku

◊ 9×9 and 16×16

◊ No same number in a row
◊ No same number in a col
◊ No same number in a sub-square

◊ Find a solution
Sudoku - Brute force

◊ Find an empty cell
◊ Enumerate a possible number to fill
◊ Mark its row, column, and subsquare

▪ Bitmasks could be used again!
Sudoku - Heuristic

- Find an empty cell
- With the **smallest** number of possible numbers
Outline

- Bitmasks for Pruning
- Search Order
- Bidirectional Search
- A* Search
Bidirectional search

◊ Find shortest path given an initial node and a target node.
◊ Two simultaneous searches
   ▪ Start → ... → an overlap! ← ... ← Target
◊ Faster!
   ▪ Suppose both searches expand a tree with branching factor $b$
   ▪ the distance from start to goal is $d$
   ▪ Each of the two searches: $O(b^{d/2})$
   ▪ A single search: $O(b^d)$
Bidirectional search

◊ Requirement: the reversed move is easy to obtain
8-puzzle

◇ 1  2  3
◇ x  4  6
◇ 7  5  8

◇ How many different layouts?
◇ Bidirectional BFS is much faster than BFS
$k$-sum

◊ Given an array $a[1..n]$ and a target sum $s$
◊ Is it possible to find $k$ numbers such that their sum is exactly $s$?

◊ $O(n^{\left\lfloor \frac{k}{2} \right\rfloor})$ is desired
Outline

- Bitmasks for Pruning
- Search Order
- Bidirectional Search
- A* Search
A* search

◊ $g(x)$ is the current cost from start to $x$
◊ Design a heuristic function $h(x)$, which estimates the cost of the cheapest path from $x$ to the goal
  ▪ problem-specific
  ▪ admissible, meaning that it never overestimates the actual cost to get to the nearest goal node
◊ $f(x) = g(x) + h(x)$
◊ Use $f(x)$ as the priority
A* search

◊ Maintain a priority queue \( Q \)

◊ Pick \( x \) with the highest priority \( f(x) \) from \( Q \)
◊ If \( x \) is the goal, \( f(x) \) is the answer
◊ Expand \( x \rightarrow y_1, y_2, \ldots, y_k \)
◊ Push all \textit{unseen} \( y \)’s into \( Q \)
Dijkstra Revisit

◊ Dijkstra's algorithm can be viewed as a special case of A*
  ▪ Pick the unseen closest node
  ▪ $g(x)$ is the current shortest distance
  ▪ $h(x) = 0$
15-puzzle

◊Goal:
◊ 1  2  3  4
◊ 5  6  7  8
◊ 9 10 11 12
◊ 13 14 15  X

◊where the only legal operation is to exchange 'x' with one of the tiles with which it shares an edge.
15-puzzle

◊ 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
◊ 5 6 7 8 5 6 7 8 5 6 7 8 5 6 7 8
◊ 9 x 10 12 9 10 x 12 9 10 11 12 9 10 11 12
◊ 13 14 11 15 13 14 11 15 13 14 x 15 13 14 15 x
◊ r-> d-> r->
15-puzzle

◊ How many different layouts?
◊ $15! = 1,307,674,368,000$
◊ Toooo large for BFS
15-puzzle

◇ Any heuristic?

◇ Sum of Manhattan Distances to their destinations
Recommended Readings

- **USACO 1.4.1 Search Techniques**
- **Bidirectional Search**
- **A* Search**
- **Exact Cover**
- **Dancing Links**
Q&A