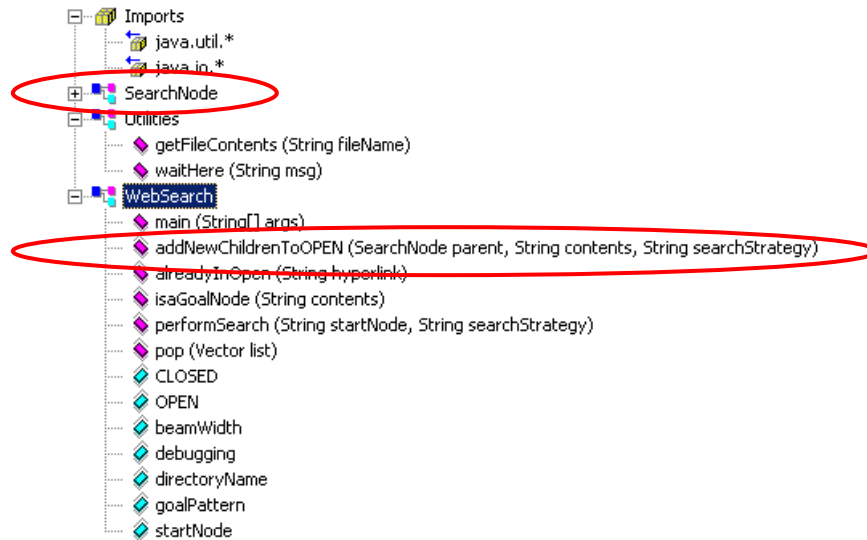


2002-09-30 Lab #1 Hints2

CS572, Fall2002, Iowa State University
Jie Bao

1- Use WebSearch frame

You only need to write code in two places:



2- The basic data structure in WebSearch.

All search algorithms needs a queue, and it's implemented by
static Vector OPEN;

(Vector can be viewed as a linear linked table in Java, and you can access, delete and add element at any position. for detail of java Class Vector, please refer to

<http://java.sun.com/products/jdk/1.2/docs/api/java/util/Vector.html>)

What you need to do in writing a searching algorithm is to choose which nodes to insert and where to insert them.

Where to insert new nodes?

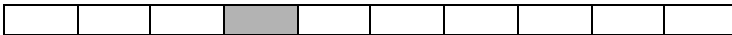
BFS



DFS



BestFS



(where the h value of new node is smaller than an existent one)

Beam



(like in BFS) Or



(like in Best First Search)

3- Beam search

Breadth first, but only keep the WIDTH best new nodes, depending on heuristic, at each level.

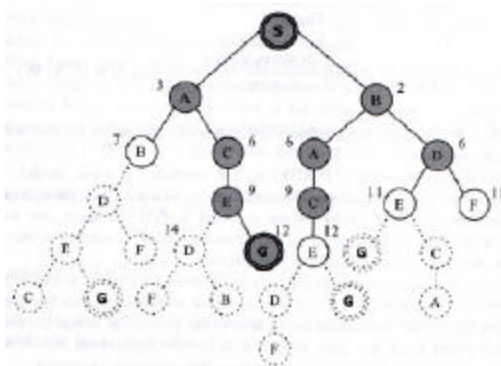
? Algorithm:

- o QUEUE <-- path containing only the root;
- o While
 - ⚡ QUEUE is not empty

✍ AND goal is not reached

DO

- ✍ remove first path from the QUEUE;
- ✍ create new paths (to all children);
- ✍ reject the new paths with loops;
- ✍ **sort new paths** heuristic
- ✍ add the first **WIDTH** new paths to the **back** of QUEUE;
- IF goal reached
 - ✍ THEN success;
 - ✍ ELSE failure;



ALGORITHM 12.5: THE BEAM SEARCH ALGORITHM

Step 1: Initialization: Put S in the OPEN list and create an initially empty CLOSE list.

Step 2: If the OPEN list is empty, exit and declare failure.

Step 3: $\forall N \in OPEN$ do

- 3a. Pop up node N in the OPEN list, remove it from the OPEN list and put it into the CLOSE list.
- 3b. If node N is a goal node, exit successfully with the solution obtained by tracing back the path along the pointers from N to S .
- 3c. Expand node N by applying a successor operator to generate the successor set $SS(N)$ of node N . Be sure to eliminate the successors, which are ancestors of N , from $SS(N)$.
- 3d. $\forall v \in SS(N)$ Create a pointer pointing to N and push it into Beam-Candidate list.

Step 4: Sort the Beam-Candidate list according to the heuristic function $f(N)$ so that the best w nodes can be pushed into the the OPEN list. Prune the rest of nodes in the Beam-Candidate list.

Step 5: Go to Step 2.

4- Avoid loop in searching!

5- What to turn in?

- ? Checklist of your files (code, report, etc.)
- ? How to run your program?
- ? How does General Search Algorithm work?
- ? Simple explain of how do you implement the Four Search Methods in your program
- ? Simple explain of how do you Structure of Search Node
- ? Simple explain of your Heuristic Function
- ? Is your heuristic *admissible*? Explain why or why not. (You're not required to write an admissible heuristic.)
- ? Running Results

Intranet1/Breadth-first

0 page1.html

1 page18.html

2 page29.html

3 page99.html

4 page50.html

Visited 91 nodes, starting @ intranet1/page1.html, using: breadth search.

intranet# breadth depth BestFS

1	91/4	58/15	19/7
3	75/6	62/6	10/6
7	56/6	12/9	27/8