1.BFS 91/4 | 88/8 | 56/6

```
daviddembp:intranets david$ java WebSearch intranet1 breadth
Path:  page50.html <- page99.html <- page29.html <- page18.html <- page1.html Visited 91 nodes,
 starting @ intranet1/page1.html, using: breadth search.

daviddembp:intranets david$ java WebSearch intranet5 breadth
Path:  page62.html <- page72.html <- page95.html <- page96.html <- page87.html <- page89.html
<- page99.html <- page40.html <- page1.html Visited 88 nodes, starting @ intranet5/page1.html,
 using: breadth search.

daviddembp:intranets david$ java WebSearch intranet7 breadth
Path:  page86.html <- page61.html <- page62.html <- page57.html <- page71.html <- page48.html
<- page1.html Visited 56 nodes, starting @ intranet7/page1.html, using: breadth search.
```

DFS 58/15 | 42/10 | 12/9

```
daviddembp:intranets david$ java WebSearch intranet1 depth
Path:  page50.html <- page83.html <- page2.html <- page79.html <- page87.html <- page93.html <- page68.html
<- page30.html <- page84.html <- page42.html <- page25.html <- page78.html <- page39.html <- page60.html
 <- page23.html <- page1.html Visited 58 nodes, starting @ intranet1/page1.html, using: depth search.

daviddembp:intranets david$ java WebSearch intranet5 depth
Path:  page62.html <- page72.html <- page95.html <- page7.html <- page48.html <- page68.html <- page97.html
<- page5.html <- page99.html <- page40.html <- page1.html Visited 42 nodes, starting @ intranet5/page1.html,
 using: depth search.

daviddembp:intranets david$ java WebSearch intranet7 depth
Path:  page86.html <- page78.html <- page11.html <- page60.html <- page39.html <- page90.html <- page57.html
 <- page71.html <- page48.html <- page1.html Visited 12 nodes, starting @ intranet7/page1.html, using: depth
 search.
```

2.a)
Heuristic function definition:

first, aim to combine the feature 1, 2, and 3 mentioned in task 2.
Because they are from weakest to strongest, I give them a coefficient : 0.2, 0.3, 0.5 (sum to 1) respectively by feeling.

Thus, $h(n) = 0.2x + 0.3y + 0.5z$

where x is the total # of QUERY words on that hyperlink page,
y is the total # of QUERY words in the hypertext associated with that hyperlink,
z is the similarity results from the goal sequence of words "QUERY1 QUERY2 QUERY3 QUERY4"
Now for z value, Define similarity by quantifying a notion of distance:

case1: only one QUERY# in hypertext, distance = 1

case2: QUERYx, QUERYy where $x + 1 = y$, distance = 2
case3: QUERYx, QUERYy where $x + 1 \neq y$, distance = 1(similar to case 1)

case4: QUERYx, QUERYy, QUERYz where $x + 2 = y + 1 = z$, distance = 3
case5: QUERYx, QUERYy, QUERYz where $x + 1 = y$ or $y + 1 = z$, distance = 2
case6: QUERYx, QUERYy, QUERYz where(else) $x + 2 \neq y + 1 \neq z$, distance = 1

case7: QUERYx, QUERYy, QUERYz, QUERYw where $x+3 = y+2 = z+1 = w$, distance =1000
(so large for we want it to represent ∞, because we find the answer, then must choose it)

case8: QUERYx, QUERYy, QUERYz, QUERYw where x+3 = y+2 = z+1 ≠ w, distance = 3
case9: QUERYx, QUERYy, QUERYz, QUERYw where x+3 ≠ y+2 = z+1 = w, distance = 3
case10: QUERYx, QUERYy, QUERYz, QUERYw where x+3 = y+2 ≠ z+1 , distance = 2
case11: QUERYx, QUERYy, QUERYz, QUERYw where x+3 ≠ y+2 = z+1 , distance = 2
case12: QUERYx, QUERYy, QUERYz, QUERYw where x+3 ≠ y+2 ≠ z+1 = w , distance = 2
case13: QUERYx, QUERYy, QUERYz, QUERYw where x+3 ≠ y+2 ≠ z+1 ≠ w , distance = 1

Now, all heuristic function setting has been done. The highest value of h(n), where n is the arc (hyperlink) coming out of the current node, will be the hyperlink that our program are gonna choose to expand next.

I do not think this is admissible heuristic. My heuristic may be taking some other path in the state space that accounts for more than the actual path cost, because there may appear the case in a intranet page that compared to other hyperlinks in that intranet page, one hyperlink with less QUERY words on that hyperlink page, less QUERY words in the hypertext associated with that hyperlink, and less similarity results, leads to the the page contains the “QUERY1 QUERY2 QUERY3 QUERY4”, which may happen.

2.c)

Best-search result: **7/5 | 30/9 | 9/7**

```
daviddembp:intranets david$ java WebSearch intranet1 best
Path:  page50.html <- page83.html <- page2.html <- page88.html <- page98.html <- page1.html Visited 7 nodes,
 starting @ intranet1/page1.html, using: best search.


daviddembp:intranets david$ java WebSearch intranet5 best
Path:  page62.html <- page72.html <- page95.html <- page35.html <- page42.html <- page19.html <- page88.html
 <- page99.html <- page40.html <- page1.html Visited 30 nodes, starting @ intranet5/page1.html, using: best
search.


daviddembp:intranets david$ java WebSearch intranet7 best
Path:  page86.html <- page61.html <- page62.html <- page57.html <- page71.html <- page48.html <- page1.html
Visited 9 nodes, starting @ intranet7/page1.html, using: best search.
```

Beam search result: **29/4 | 70/8 | 37/6**

```
daviddembp:intranets david$ java WebSearch intranet1 beam
Path:  page50.html <- page99.html <- page88.html <- page98.html <- page1.html Visited 29 nodes, starting @ i
ntranet1/page1.html, using: beam search.


daviddembp:intranets david$ java WebSearch intranet5 beam
Path:  page62.html <- page72.html <- page95.html <- page96.html <- page87.html <- page89.html <- page99.html
 <- page40.html <- page1.html Visited 70 nodes, starting @ intranet5/page1.html, using: beam search.


daviddembp:intranets david$ java WebSearch intranet7 beam
Path:  page86.html <- page61.html <- page62.html <- page57.html <- page71.html <- page48.html <- page1.html
Visited 37 nodes, starting @ intranet7/page1.html, using: beam search.
```

Thus, comparing to BFS,DFS results, clearly my heuristic formula works well which reduce more nodes needed to visit to find a solution path, although the path may not be the shortest.

3)using jsoup
haven't finished
http://www.cs.iastate.edu
Query words: Des Moines