Massachusetts Institute of Technology

6.034 Artificial Intelligence

Solutions # **4** 6034 Item # **6**

Problem 1

The search will begin to loop back, visiting places it has already been:

```
(step-by-step '(mit))
The path (mit) can be extended to get (mt-holyoke mit)
Continue? [y or n]: y
The path (mt-holyoke mit) can be extended to get (smith mt-holyoke mit)
Continue? [y or n]: y
The path (smith mt-holyoke mit) can be extended to get (mt-holyoke smith mt-holyoke mit)
Continue? [y or n]: n
```

Here's one way to fix the problem:

```
(define (extend-path path)
  "
  Purpose:  Extend a path to the neighbors of the terminal host
  Returns:  A list of extended paths
  "
; (format #t "~%Extending the path ~a." (reverse-path path))
  (map (lambda (nexthost)
         (cons nexthost path))
       (remove-loops (get-neighboring-hosts (first-host path))
                     path)))

(define (remove-loops extensors path)
  ;;; Takes a list of potential extensors to a path, and a path
  ;;; Returns a possibly reduced list of extensors, removing any of them
  ;;; that are already on the path.
  (cond ((null? extensors) '())
        ((member (first extensors) path)
         ;;; been there, done that, ignore it
         (remove-loops (rest extensors) path))
        (else
         (cons (first extensors) (remove-loops (rest extensors) path)))))
```

With this change we get:

```
(step-by-step '(mit))
The path (mit) can be extended to get (mt-holyoke mit)
Continue? [y or n]: y
The path (mt-holyoke mit) can be extended to get (smith mt-holyoke mit)
Continue? [y or n]: y
The path (smith mt-holyoke mit) can be extended to get (amherst smith mt-holyoke mit)
Continue? [y or n]: y
The path (amherst smith mt-holyoke mit) can be extended to get (dartmouth amherst smith
  mt-holyoke mit)
Continue? [y or n]: y
The path (dartmouth amherst smith mt-holyoke mit) cannot be extended any further.
```

Problem 2

```
(define (depth-first originhost destnhost)
  "
  Purpose:      Perform depth-first search
  Remark:       Uses generalized search function
  "
  (define (combine-paths new-paths old-paths finish)
    "
    Purpose:    Define a queue constructor specific to depth-first search.
                Note that the finish argument does not need to be used.
    "
    (append new-paths old-paths)
    )

  ;; Fire up generalized search using queue constructor defined above:
  (generalized-search originhost destnhost (make-finish-test destnhost)
                      combine-paths 1 extend-path))
```

```
(depth-first 'mit 'cmu)
Extending the path (mit).
Extending the path (mit mt-holyoke).
Extending the path (mit mt-holyoke smith).
Extending the path (mit mt-holyoke smith amherst).
Extending the path (mit mt-holyoke smith amherst dartmouth).
Extending the path (mit mt-holyoke smith amherst bu).
Extending the path (mit mt-holyoke smith amherst bu hahvud).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst penn).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst
  penn princeton).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst
  penn princeton ga-tech).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst
  penn princeton ga-tech tenn).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst
  penn princeton ga-tech tenn urbana).
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst
  penn princeton ga-tech chicago).
```

```
Extending the path (mit mt-holyoke smith amherst bu hahvud wellesley babson umass-amherst
  penn princeton ga-tech chicago columbia).
Final path length is 15
Proceed from mit to mt-holyoke
then from mt-holyoke to smith
then from smith to amherst
then from amherst to bu
then from bu to hahvud
then from hahvud to wellesley
then from wellesley to babson
then from babson to umass-amherst
then from umass-amherst to penn
then from penn to princeton
then from princeton to ga-tech
then from ga-tech to chicago
then from chicago to columbia
then from columbia to cmu.
```

Problem 3

```
(define (breadth-first originhost destnhost)
  "
  Purpose:      Perform breadth-first search
  Remark:       Uses generalized search function
  "
  (define (combine-paths new-paths old-paths finish)
    (append old-paths new-paths)
    )

  ;; Fire up generalized search using queue constructor defined above:
  (generalized-search originhost destnhost (make-finish-test destnhost)
                      combine-paths 1 extend-path))
```

```
(breadth-first 'mit 'cmu)
Extending the path (mit).
Extending the path (mit mt-holyoke).
Extending the path (mit amherst).
Extending the path (mit bu).
Extending the path (mit hahvud).
Extending the path (mit wellesley).
Extending the path (mit mt-holyoke smith).
Extending the path (mit mt-holyoke amherst).
Extending the path (mit mt-holyoke bu).
Extending the path (mit amherst dartmouth).
Extending the path (mit amherst smith).
Extending the path (mit amherst mt-holyoke).
Extending the path (mit amherst bu).
Extending the path (mit amherst hahvud).
Extending the path (mit bu mt-holyoke).
Extending the path (mit bu amherst).
Extending the path (mit bu hahvud).
Extending the path (mit bu wellesley).
Extending the path (mit hahvud amherst).
```

```
[ trace shortened to save trees]

Extending the path (mit wellesley princeton penn tenn babson).
Extending the path (mit wellesley princeton penn tenn ga-tech).
Extending the path (mit wellesley princeton penn tenn urbana).
Extending the path (mit wellesley princeton penn urbana tenn).
Extending the path (mit wellesley princeton ga-tech tenn babson).
Extending the path (mit wellesley princeton ga-tech tenn penn).
Extending the path (mit wellesley princeton ga-tech tenn urbana).
Extending the path (mit wellesley princeton ga-tech chicago columbia).
Final path length is 6
Proceed from mit to wellesley
then from wellesley to princeton
then from princeton to ga-tech
then from ga-tech to chicago
then from chicago to cmu.
```

Problem 4

```
;;; here are the data abstractions for a path that is a list of hosts with the
;;; path length at the front.

(define empty-path? null?)
(define first-host second)
(define second-host third)
(define (remaining-hosts path) (rest (rest path)))
(define reverse-path reverse)
(define (construct-path host) (list 0 host))

(define path-length first)

;;; and we need one new one
(define path-hosts rest)

;; need a small change to goal-action so it prints properly
(define (goal-action queue)
  (format #t "~%Final path length is ~s" (path-length (first-path queue)))
  (describe-path (reverse-path (path-hosts (first-path queue)))))

;;; now we can create the branch and bound searcher

(define (branch-and-bound originhost destnhost)
  (define (extend-path-pl path)
  ;;; this version creates paths that have their path-length at the front
  ;;; it builds on the previously debugged extend-path
    (format #t "~%Extending the path ~a." (reverse-path path))
    (map (lambda (nexthost)
           (cons (+ (path-length path)
                    (straight-line-distance nexthost (first-host path)))
                 (cons nexthost (path-hosts path))))
         (remove-loops (get-neighboring-hosts (first-host path))
                       path)))

  (define (combine-paths-pl new old finish)
  ;;; sort the combined old and new list according to path-length
    (sort (append new old)
          (lambda (p1 p2) (< (path-length p1) (path-length p2)))))

  (generalized-search originhost destnhost (make-finish-test destnhost)
                      combine-paths-pl 1 extend-path-pl))


(branch-and-bound 'dartmouth 'wellesley)
Extending the path (dartmouth 0).
Extending the path (dartmouth amherst 85.).
Extending the path (dartmouth amherst smith 163.).
Extending the path (dartmouth amherst mt-holyoke 165.).
Extending the path (dartmouth amherst bu 166.).
Extending the path (dartmouth amherst hahvud 167.).
Extending the path (dartmouth amherst mit 178.).
```

```
Extending the path (dartmouth amherst bu hahvud 188.).
Extending the path (dartmouth amherst hahvud bu 189.).
Extending the path (dartmouth amherst bu mit 210.).
Extending the path (dartmouth amherst mit bc 220.).
Extending the path (dartmouth amherst smith mt-holyoke 220.).
Extending the path (dartmouth amherst mt-holyoke smith 222.).
Extending the path (dartmouth amherst mit bu 222.).
Extending the path (dartmouth amherst mt-holyoke mit 225.).
Extending the path (dartmouth amherst hahvud bu mit 233.).
Extending 1the path (dartmouth amherst hahvud mit 234.).
Extending the path (dartmouth amherst mit mt-holyoke 238.).
Final path length is 239.
Proceed from dartmouth to amherst
then from amherst to hahvud
then from hahvud to wellesley.
;Value 29: ((239. wellesley hahvud amherst dartmouth))
```

The code always presses forward from the minimum path length.

As one example making it evident that we are not using the dynamic programming principle, the code finds a path from Dartmouth to BU that is 166 units long, then finds one that is 189 units long.

Problem 5

```
(define (a-star originhost destnhost)
  (define (extend-path-pl path)
  ;;; this version creates paths that have their path-length at the front
  ;;; but it uses distance so far as well as estimated remaining distance
    (format #t "~%Extending the path ~a." (reverse-path path))
    (map (lambda (nexthost)
           (cons (+ (path-length path)
                    (straight-line-distance nexthost (first-host path))
                    ;;; now we have to remove the old estimate of remaining dist
                    ;;; (but not the for the first node; ugly but effective
                    (if (null? (remaining-hosts path))
                        0
                        (- (straight-line-distance (first-host path) destnhost)))
                    ;;; and put on the new onw
                    (straight-line-distance nexthost destnhost))
                 (cons nexthost (path-hosts path))))
         (remove-loops (get-neighboring-hosts (first-host path))
                       path)))

  (define (combine-paths-pl new old finish)
  ;;; sort the combined old and new list according to path-length
    (sort (append new old)
          (lambda (p1 p2) (< (path-length p1) (path-length p2)))))

  (generalized-search originhost destnhost (make-finish-test destnhost)
                      combine-paths-pl 1 extend-path-pl))

(a-star 'dartmouth 'wellesley)
Extending the path (dartmouth 0).
Extending the path (dartmouth amherst 237.).
Extending the path (dartmouth amherst bu 239.).
Extending the path (dartmouth amherst hahvud 239.).
Final path length is 239.
Proceed from dartmouth to amherst
then from amherst to hahvud
then from hahvud to wellesley.
```