CS1110 4 November 2010

Developing array algorithms. Reading: 8.3..8.5

Prelim Tuesday, 7:30PM. Olin 155 and 255
Review session, Sunday 1–3. Phillips 101
Handout describes what will be covered.

have iclickers ready for quiz

1

Two-dimensional arrays

	0	1	2	3	b.length
b	5	4	7	3	

one-dimensional array

d	0	1	2	3
0	5	4	7	3
1	4	8	9	7
2	5	1	2	3

rectangular array: 5 rows and 4 columns

Type of d is **int**[][] ("**int** array array", "an array of **int** arrays")

To declare variable d:
int d[][].

To create a new array and assign it to d:
d= **new int**[3][4];

To reference element at row r column c:
d[r][c]

number of rows

number of cols

2

A 2-dimensional array b P00 P01 P02 P03
P10 P11 P12 P13
P20 P21 P22 P23

Same array in row-major order:

c P00 P01 P02 P03 P10 P11 P12 P13 P20 P21 P22 P23

You can see that

b[0][j] is same as c[0 * (no of columns) + j]

b[1][j] is same as c[1 * (no of columns) + j]

and in general:

b[i][j] is same as c[i * (no of columns) + j]

3

Pixel (picture element): 4 components, each in 0..255

Contains: alpha component (we never change it)
red component r
green component g
blue component b

```
/** DM provides functions for extracting components of a pixel. */
public static final DirectColorModel DM=
        (DirectColorModel) ColorModel.getRGBdefault();
```

Procedure invert has in it: DM.getRed(pixel);

8 bits	8 bits	8 bits	8 bits
alpha	red	green	blue

all 4 components fit into an **int**

(alpha << 24) | (red << 16) | (green << 8) | blue)

4

Dutch National Flag

pre b	0	?	n

contains red, white, blue balls

Permute b[0..n-1] to truthify:

post b	0	reds	whites	blues	n

Generalize the pre- and post-condition to get a loop invariant.

5

Partition algorithm: Given an array b[h..k] with some value x in b[h]:

P: b	h	x	?	k

Swap elements of b[h..k] and store in j to truthify P:

Q: b	h	<= x	j	x	>= x	k

change: b h k: 3 5 4 1 6 2 3 8 1

into b h j k: 1 2 1 3 5 4 6 3 8

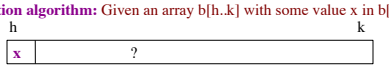
or b h j k: 1 2 3 1 3 4 5 6 8

x is called the pivot value.

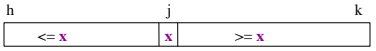
x is not a program variable; x just denotes the value initially in b[h].

6

Partition algorithm: Given an array b[h..k] with some value x in b[h]:



Swap elements of b[h..k] and store in j to truthify Q:



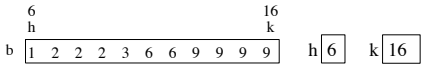
You generalize P and Q to create a loop invariant

x is called the pivot value.

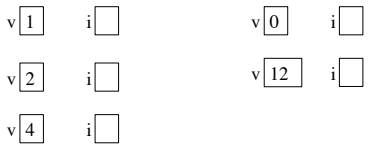
x is not a program variable; x just denotes the value initially in b[h].

7

Binary search: Vague spec: Look for v in **sorted** array segment b[h..k].



Store the index of v in variable i

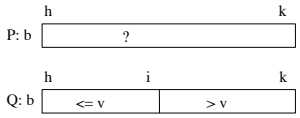


8

Binary search: Vague spec: Look for v in **sorted** array segment b[h..k].

Precondition P: b[h..k] is sorted (in ascending order).
Store in i to truthify:
Postcondition Q: b[h..i] <= v and v < b[i+1..k]

Below, the array is in non-descending order:

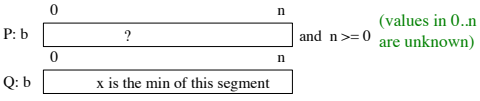


Called binary search because each iteration of the loop cuts the array segment still to be processed in half

9

Invariant as picture: Generalizing pre- and post-condition

Finding the minimum of an array. Given array b satisfying precondition P, store a value in x to truthify postcondition Q:

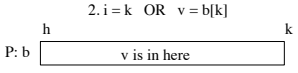


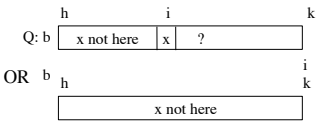
10

Linear search

Vague spec.: Find first occurrence of v in b[h..k-1].
Better spec.: Store an integer in i to truthify postcondition Q:

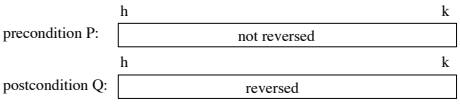
Q: 1. v is not in b[h..i-1]
2. i = k OR v = b[k]





11

Reversal: Reverse the elements of array segment b[h..k].



Change: b 1 2 3 4 5 6 7 8 9 9 9 9

into b 9 9 9 9 8 7 6 5 4 3 2 1

12