CS50
Walkthrough #3
Marta Bralic
mbralic@fas.harvard.edu
Agenda

- generate.c
- Makefile
- find.c
  - search
  - sort
- fifteen.c
  - distribution code
    - gdb
  - init
  - draw
  - move
  - won
generate.c

- what does it do?
- comments!
Makefile

- what does it do?
Search

- Linear Search
- Re-implement as binary!
  - why?
- 2 main ways
  - iterative
  - recursive
Binary Search: Iterative

Go to middle
  if \( k < \) value at middle
    search for \( k \) between first and the one before the middle
  if \( k > \) value at middle
    search for \( k \) between one after the middle and last
  if \( k = \) value at middle
    return true

If you haven’t found \( k \) after this loop, return false
Binary Search: Recursive

search(array, first, last, k)
  if first > last
    return false
  else if k < array[middle]
    search(array, first, middle-1, k)
  else
    ...

...
Sort: Bubble

repeat n times
  for each element i
    if i and its neighbor are out of order, swap them

what is the running time?
Bubble sort: improvement

if you’ve made no swaps
stop sorting – you’re done

what is the running time?
Selection Sort

\( i = 0 \)

repeat \( n \) times
  
  find the smallest value (s) between \( i \) and the end
  
  swap s with element at location \( i \)
  
  \( i++ \)
fifteen.c

- distribution code → main
- gdb
init()

- two dimensional array to store board values
  - what type are these values?
  - how do we initialize them?

- don't forget!
  - swap tiles for even d
  - initialize the empty tile
draw()

- what tool do we use to draw?
- how do we access the values we need?
  - where are they stored?
move()

- bool

- check for a blank space
  - if possible, swap
  - do not check for a blank outside the bounds of the array
won()

- bool
- several ways
  - check that numbers are sorted from least to greatest
  - use a counter variable to ensure each value is in place
  - other ways?
Questions?

Please email me feedback: mbralic@fas.harvard.edu