

lecture 34

graph applications

- garbage collection
- Google page rank

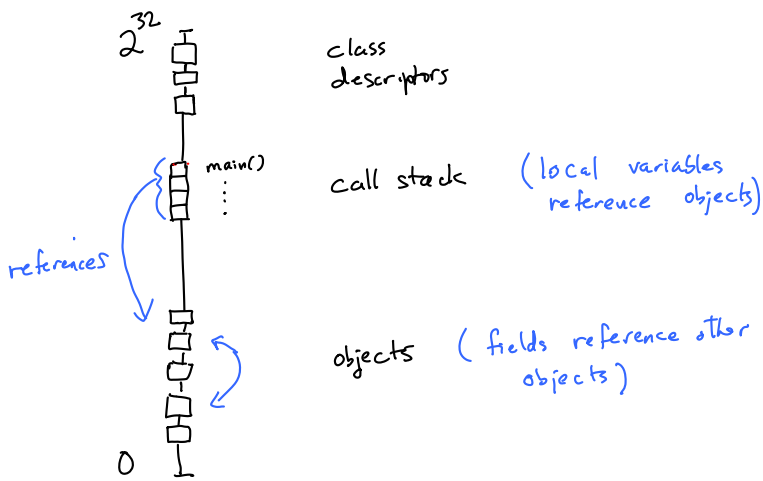
Garbage Collection

```
myDog = new Beagle("Bob")
myDog = new Terrier("Tim")
```

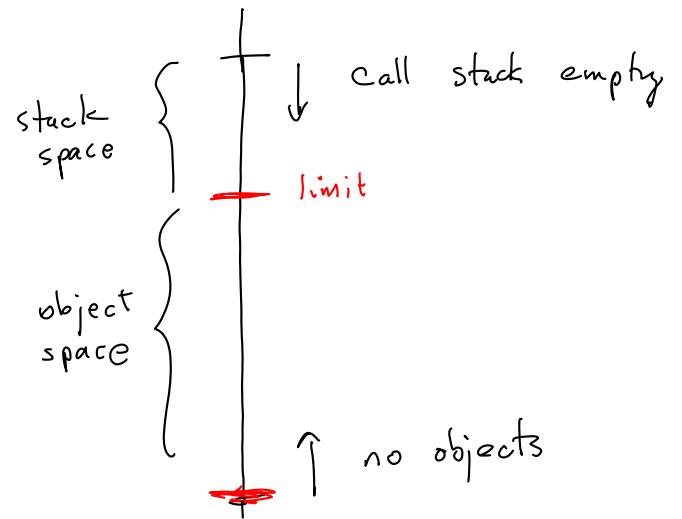
Nothing references Bob
Bob is wasting memory.

Bob is garbage.

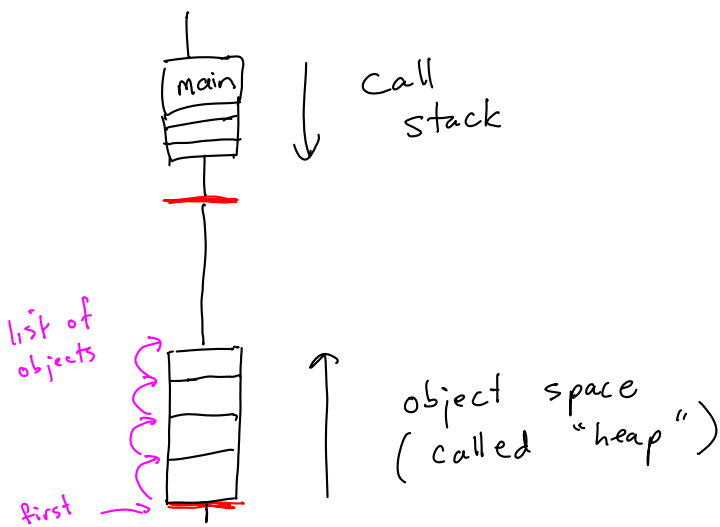
Memory of Java Virtual Machine



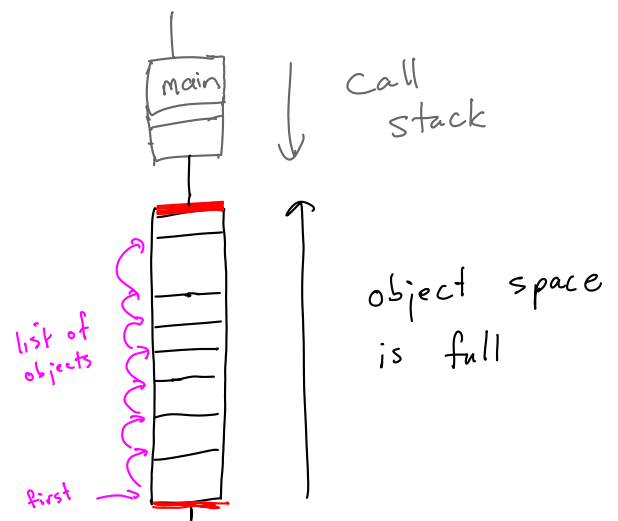
When the program starts



As program runs ...



Eventually

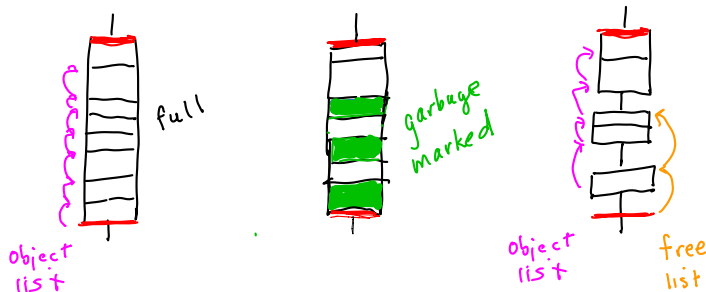


Object Space ("Heap") is full.
What to do?

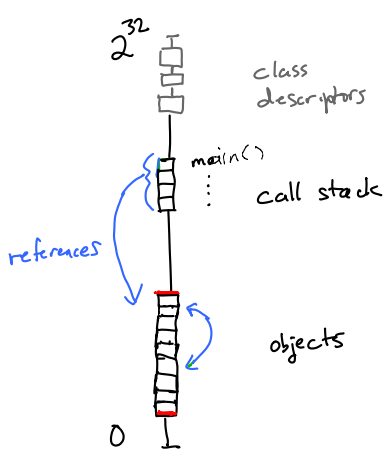
- let program crash
- reuse space we don't need anymore (garbage collection)

Garbage collection: two phases

- ① identify ("mark") → ② remove ("sweep")



① How to identify garbage?

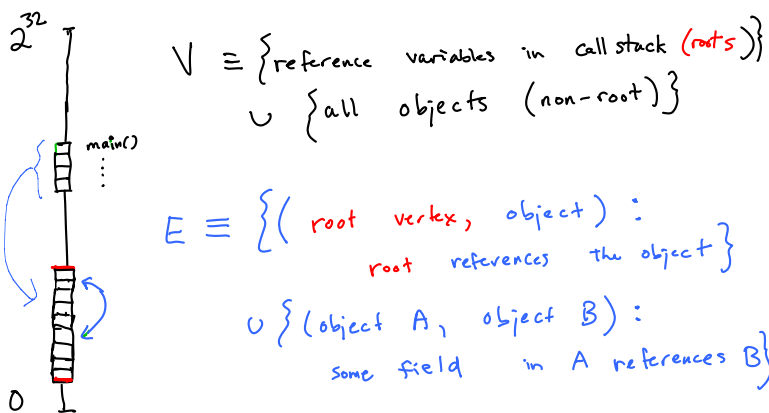


An object is "live" (not garbage) if it is **referenced**

- by a variable on the stack
- or
- by another "live" object (recursion)

① How to identify garbage?

Consider a graph:



① How to identify garbage objects?

// Let each object have a visited bit
// (also called a "mark" bit)

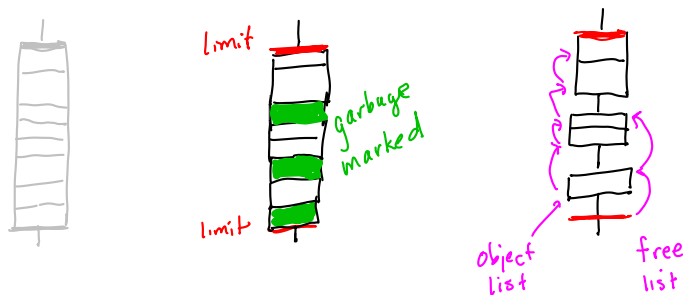
Initialize each object.visited to false

For each root vertex r ,
traverse the graph, visiting all
objects (reachable from r)

// if object is not visited by any of these
// traversals, then object is garbage

How to remove garbage?

- ① identify ("mark") → ② remove ("sweep") (adjust lists below)



Then, continue program execution

- new objects can be added, wherever there is a gap big enough (adjust the object vs. free lists)
- garbage collection needed again when there is no gap big enough for the new object
- program needs to stop (temporarily) to do garbage collection \rightarrow not good for important real time stuff eg. video game

lecture 34

graph applications

- garbage collection
- Google page rank

What does Google do?

- download all the reachable pages on web (web crawler)

- build graph $V = \{\text{web pages}\}$
 $E = \{(p_1, p_2) \text{ if } p_1 \text{ has link to } p_2\}$

- analyse the "importance" of each page ("page rank")

Google's Page Rank

How "important" is a web page v ?

Recursive concept of importance

- You are important if important people think you are important.
- Concrete example: in academic research, a published paper is important if it is cited by many important published papers. (especially those important papers that cite relatively few papers)

Google's Page Rank

How important is a web page v ?



- how many pages w point to v ?
- how important are each of these pages w (that point to v)?
- how many pages do each of these pages w point to?

How important is a web page v ?



Define $R(w)$ to be the pagerank of w .

Let $N_{out}(w)$ be the out-degree of w , i.e. the number of outgoing edges

$$R(v) = \sum_{(w,v) \in E} \frac{R(w)}{N_{out}(w)}$$

How to solve it?

$$R(v) = \sum_{(w,v) \in E} \frac{R(w)}{N_{\text{out}}(w)}$$

- recursion, but no base case
- $R^0(v) = \text{constant}$ (initial guess)
- $R^{k+1}(v) = \sum_{(w,v) \in E} \frac{R^k(w)}{N_{\text{out}}(w)}$
and iterate until convergence

What does Google do?

- offline (done in advance)
- download all the reachable pages on web
 - build graph
 - analyse the "importance" of each page ("page rank")

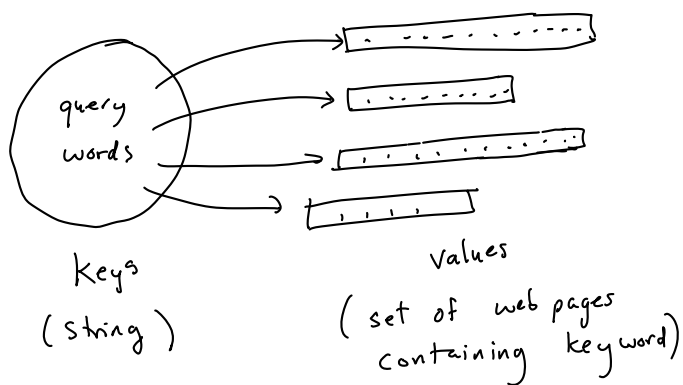
What does a user do?

- online
- enter query words and ask Google search engine for list of pages containing these words

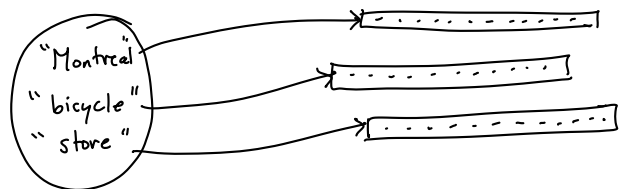
What does Google do?

- deliver pages ordered by pagerank

To answer a user's query, Google needs to index words



Example



- Then, compute the intersection of these three sets, giving web pages that contain all three keywords
- Then, sort by pagerank $R()$

For more details, see:

<http://infolab.stanford.edu/~sergey>

for Sergey Brin's old web page