Problem Solving Strategies

Problem Solving Strategies

What are examples of problems that can be solved with the following methods?

- Generate and test
- Divide and Conquer
- Simulation
- Approximation
- Adaptation

Generate and Test

- Generate candidate solutions
- Test if they are valid

Careful if infinite:

- have a counter for loops, exist when it meets a threshold

General structure:

```
int niters = 0;
while (ninters < THRESHOLD) {
        int next = generate();
        if (check(next)) break;
        ninters++;
}
```

Generate and Test

- Generate candidate solutions
- Test if they are valid

```
int
main(int argc, char *argv[]) {
        int n;
        printf("Enter a number n: ");
        scanf("%d", &n);
        if (isprime(n)) {
                printf("%d is a prime number\n", n);
        } else {
                printf("%d is not a prime number\n", n);
        }
        printf("The next prime is : %d\n", nextprime(n));
        return 0;
}

/* Determine whether n is prime. */
int
isprime(int n) {
        int divisor;
        if (n<2) {
                return 0;
        }
        for (divisor=2; divisor*divisor<=n; divisor++) {
               if (n%divisor==0) {
                       /* factor found, so can't be prime */
                       return 0;
                }
        }
        /* no factors, so must be prime */
        return 1;
}

int
nextprime(int n) {
        n = n+1;
        while (!isprime(n)) {
                n = n+1;
        }
        return n;
}
```

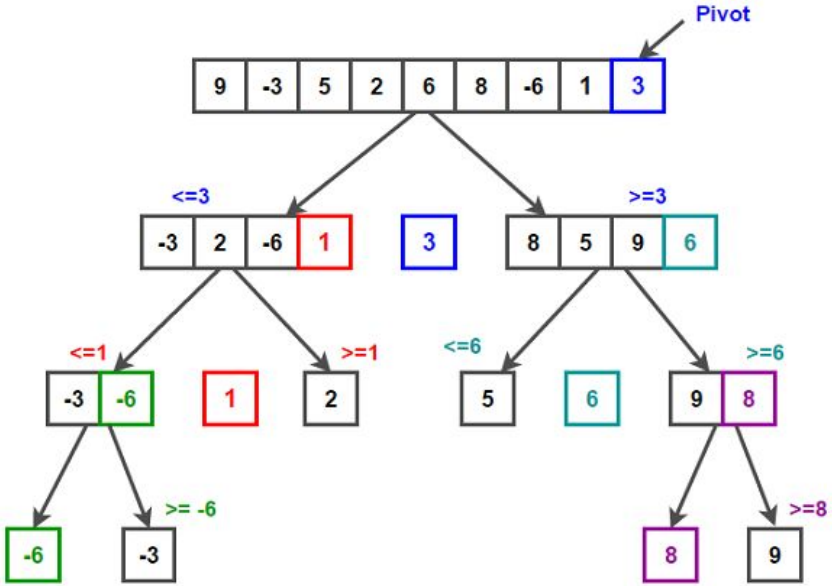
Test if prime

Generate the next prime

Divide and Conquer

- Break problem into smaller instances (**divide**)
- solve the instances (recursively?) (**conquer**)
- **combine solutions** to create a solution to the original problem

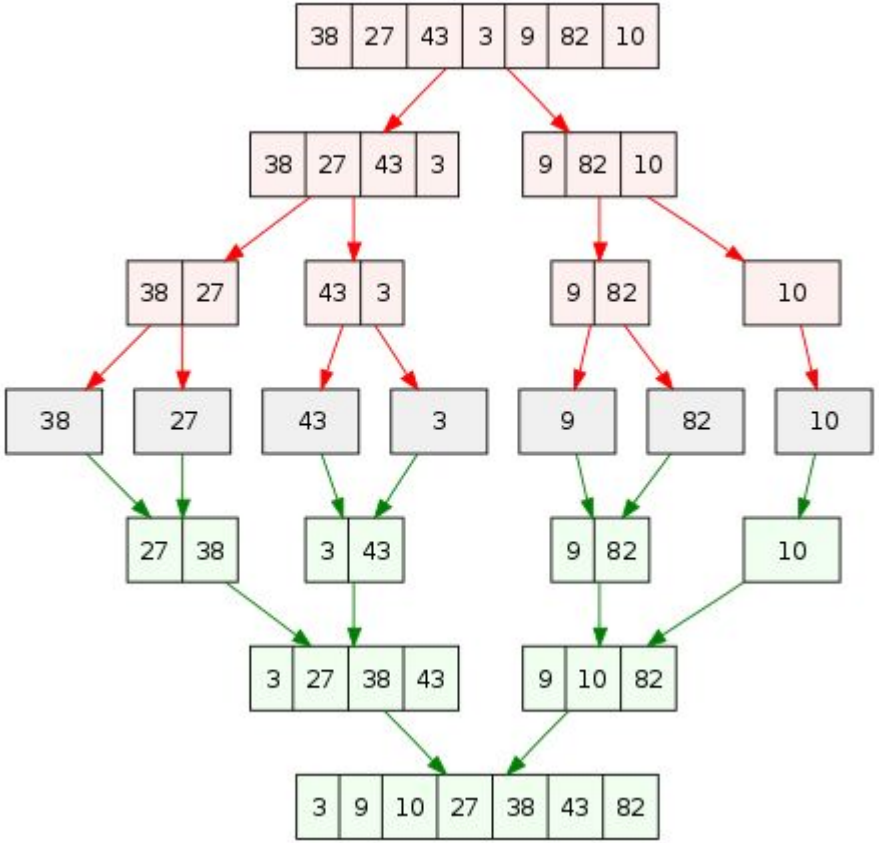
mergesort, quicksort, subset sums, hanoi, binary search

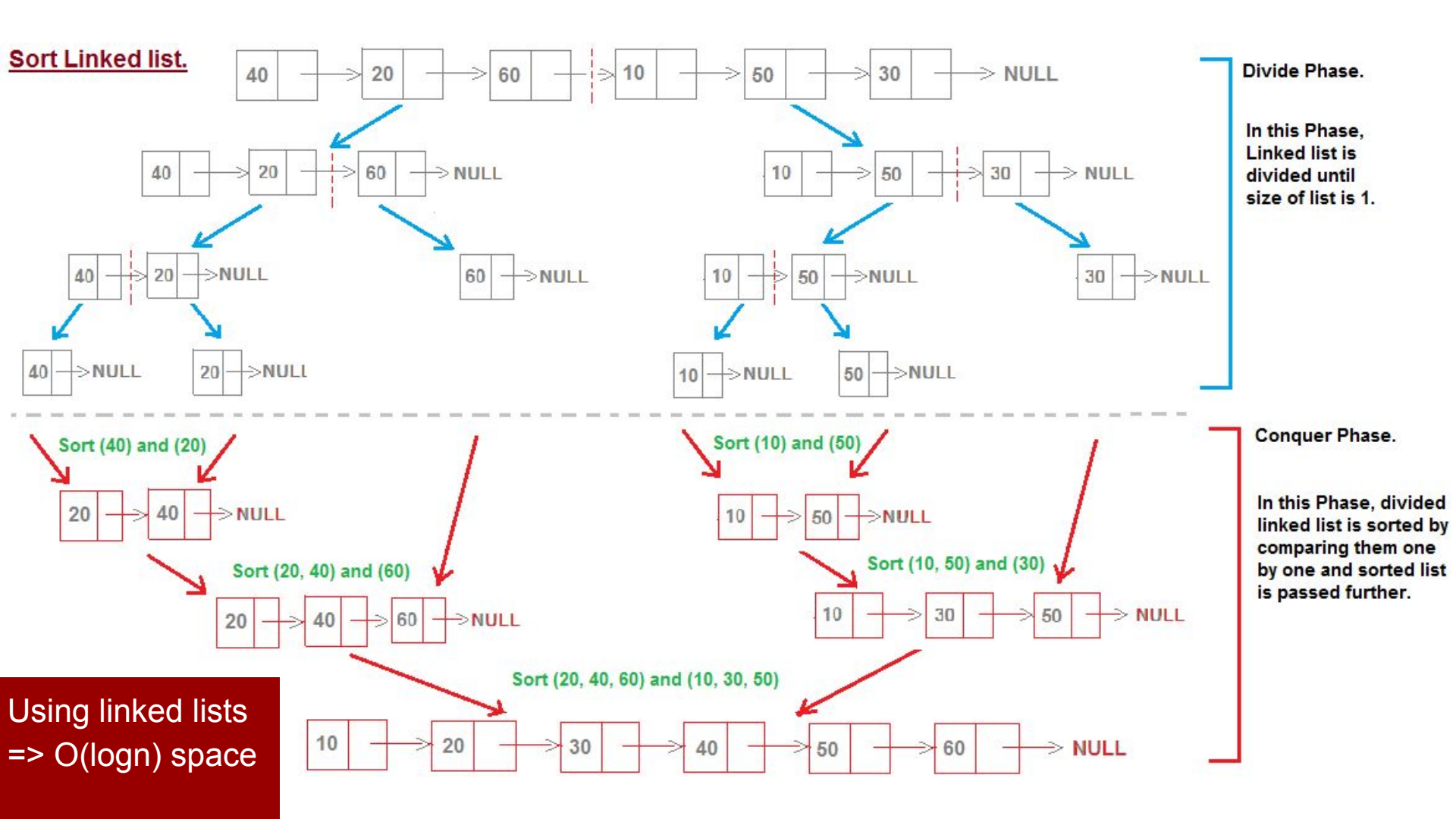


Mergesort

O(n log n) worst / best case

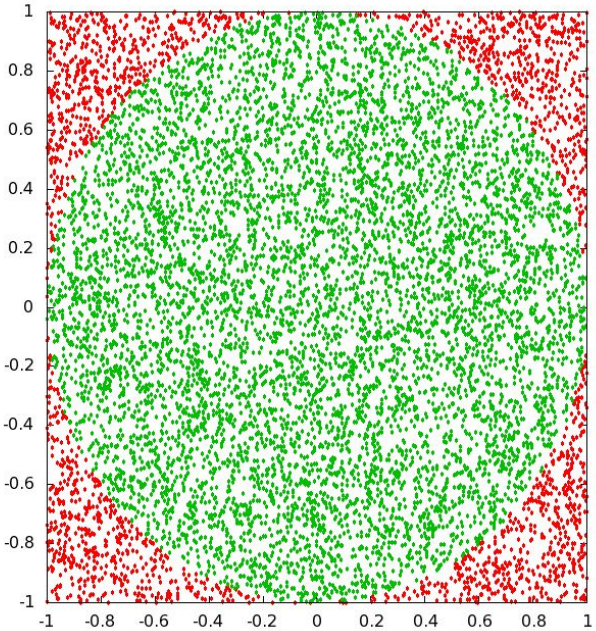
but O(n) space.




Sort Linked list.
40 20 60 10 50 30 NULL
40 20 60 NULL
10 50 30 NULL
40 20 NULL
60 NULL
10 50 NULL
30 NULL
40 NULL
20 NULL
10 NULL
50 NULL
Divide Phase.
In this Phase, Linked list is divided until size of list is 1.
Sort (40) and (20)
20 40 NULL
Sort (10) and (50)
10 50 NULL
Sort (20, 40) and (60)
20 40 60 NULL
Sort (10, 50) and (30)
10 30 50 NULL
Sort (20, 40, 60) and (10, 30, 50)
10 20 30 40 50 60 NULL
Conquer Phase.
In this Phase, divided linked list is sorted by comparing them one by one and sorted list is passed further.
Using linked lists
=> O(logn) space

Simulation (Monte Carlo)

- Use pseudo-random number generation to allow modeling of a physical system.
- Mimics physical phenomena




```
#define SEED 7716977

int inside(double, double);

int
main(int argc, char *argv[]) {
        int num_in, steps, i;
        double x, y;
        srand(SEED);
        for (steps=1; steps <= 1000000; steps=steps*10) {
                num_in=0;
                for (i=0; i<steps; i++) {
                        x = rand()/(1.0+RAND_MAX);
                        y = rand()/(1.0+RAND_MAX);
                        num_in = num_in + inside(x,y);
                }
                printf("steps = %7d, num_in = %8d, ratio = %.6f\n",
                        steps, num_in, (double)num_in/steps);
        }
        return 0;
}

int
inside(double x, double y) {
        return ((1-x)<=y && x*x+y*y<=1.0);
}
```

Area of a section of a circle

r = 1 pi * r^2 = pi <- approximation for pi

Random number generation

PRNG - pseudo-random number generation – computers can't be random, they're following an algorithm!

srand(time(NULL)) seed the random number generator (time(NULL) needs <time.h>)
the next 'random number' depends on a seed (think: starting number)

rand() generate a random number (between 0 and RAND_MAX>=32767)

6 sided dice? `d6 = 1 + rand() % 6;`
Picks a random number between 1 and 6

Toss 'H' (heads) or 'T' (tails):

```
char *coin = “HT”;
char toss = coin[rand()%2];
```

```
#include <stdlib.h>
#include <time.h>
srand(time(NULL)); // must remember to seed the RNG!
rand() // returns a large integer
rand() % n // returns integer between 0 and n-1
```

x2 + x + 5 = 0

Approximation

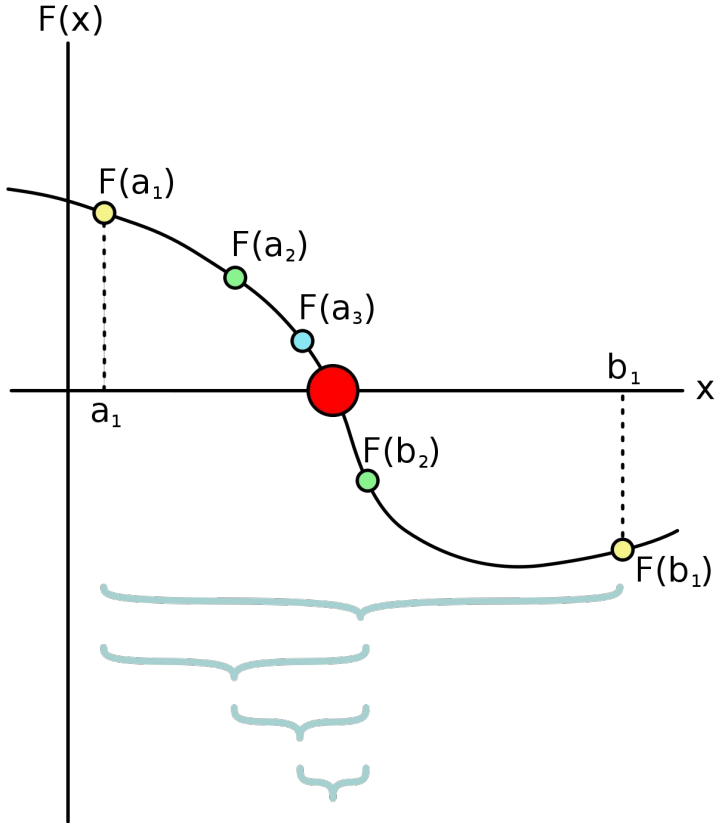
- *Solve a simpler problem that approximates the original problem, also estimating the error*

Error? Here we have **real numbers** - we won't be exact, we will be within EPSILON error.

Examples

- Integration
- curve fitting
- root finding
- DE solutions (Bisection method)

https://people.eng.unimelb.edu.au/ammoffat/ppsaa/c/bisection.c



bisection method - finding root of polynomial

Problem Solving Strategies

What are examples of problems that can be solved with the following methods?

- Generate and test >> find primes.
  Loop through solution space, test each candidate
- Divide and Conquer >> Tower of Hanoi, Subset sums, Sorting (Quicksort, Mergesort)
  (Recursively) *Break problem into subproblems. Solve the sub problems.*
  Combine solutions to create a solution to the original problem.
- Simulation >> Gambling, Poker hands
  Monte carlo methods -- PRNG models a physical system
- Approximation >> Integration, curve fitting, root finding, DE solutions (Bisection method)
  Solve a simpler problem that approximates the original problem, also estimating the error
- Adaptation
  - Modify the solution / approach used for another problem