

- There are 50 4-person suites for 200 people moving in
- any member can be a roommate with another
- Each student has a rating compatible with another student (it is symmetrical and with 4 it is pairwise)

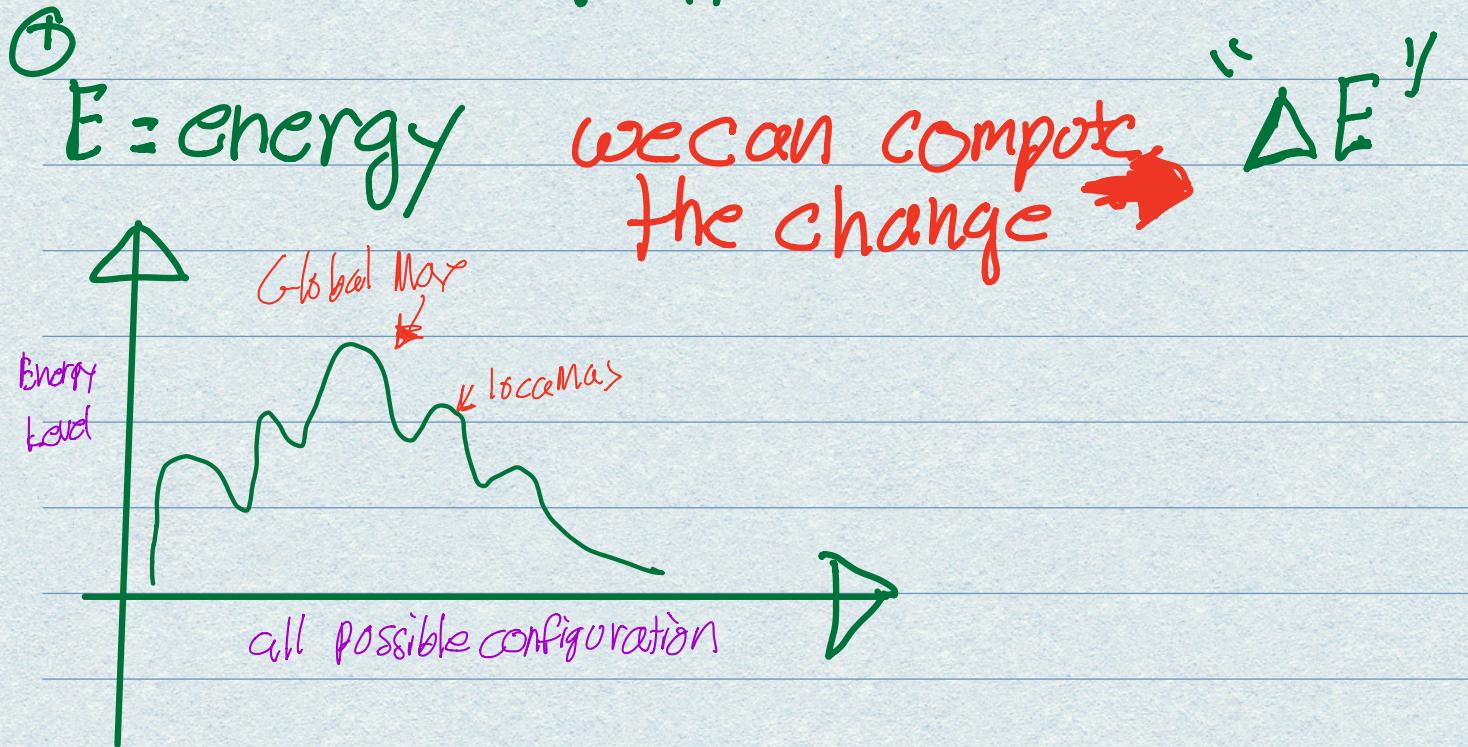
START OFF WITH:
 FCFS \rightarrow first 4 students in room 1, etc

*Fitness is the total of the pairwise
 - low Fitness score = Better compatibility

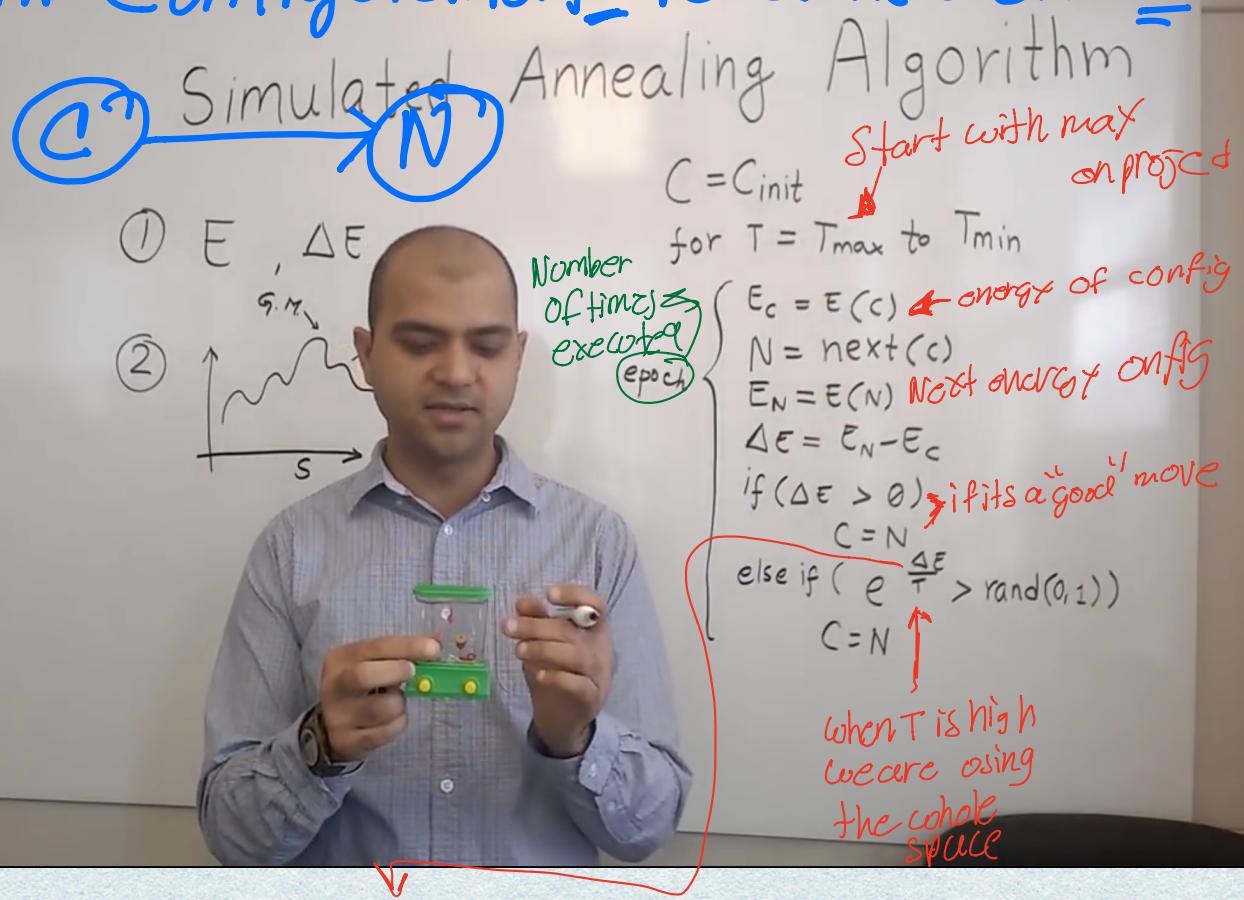
In order to solve a problem like simulated annealing, every problem needs a way to define how good a configuration is.

This is called the "Energy"

of the Solution



- We need a Mechanism to change the current configurations to a new one N



The change in delta also influences it. If the change is high, there is a low probability to accept the move. If the change is small, there is a high probability to accept the move.

54)

86

84

26

$[0] + [1]$

$[0] + [2]$

$[0] + [3]$

$[1] + [2]$

A screenshot of a mobile application interface. At the top, there is a navigation bar with a back arrow, the text "0:16 PM Tue Mar 16", and a location indicator "Kansas City - Central Hyde Park Sunday 3:49 PM". On the right side of the nav bar are icons for a trash can, a heart, and a camera, followed by "Edit All Photos". Below the nav bar, there is a "LIVE" indicator. The main area of the screen shows a code editor with the following C++ code:

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7
8 int main()
9 {
10     struct house{
11         //list ok numbers
12         //position num
13     };
14
15     house h1[2][2];
16     h1[0][1].home=5;
17     h1[2][2].home=6;
18     int cool =h1[0][1].home+h1[2][2].home;
19     cout<<cool;
20 }
```

Below the code editor is a "Get URL" button and three tabs: "options", "compilation", and "execution". The "compilation" tab is selected. At the bottom of the screen, there is a horizontal bar with many small thumbnail images of other screens or documents.

Assign People to rooms idea

Initializing multiple keys at once C++
Room 0 → Room 49

Ideas: hash of each row

#0	#1	#2	
0	54	86	84
54	0	67	26
86	67	0	39
84	26	39	0

*

We are going to select 2 student
1 for AB
1 for AC

54

86

84

26

$[0] + [1]$

$[0] + [2]$

$[0] + [3]$

$[1] + [2]$

39

$[2] + [3]$

rand 0

rand 1

rand 2

rand 4

Plan

- create 2 dimensional array
 - fill it in
- (AB, AC, AD, DC, BB
CB)
random A 0, 1
random B [A] [B]
random C [C] [D]
random D

May store in hash

So filling the rooms initially does not use simulated annealing. We use simulated annealing with the two change choices?

Best Book is WISNU ANGGORO

C++ Data Structures and algos

Paclut

Each person must be a node to prevent same person twice

New plan

build a struct
each struct has a
list of compet
and person num

outside for loop

add to

inside add |

make a list of list
of Nodes

56 rooms

4 people to each room

200 numbers

0

0

0 → 20

0, 54, 86, 89

While true

Search

Person new location

Int newroom¹ newroom
newperson¹ new location

↓ newroom² new room
newperson² new local

Person1 = was at [new¹] [new²]
Person1.identify = was [] [].identity

Person2 = was at [] []

Person2 = was [] [] - identify

was [new room] [new] = P2

was [new¹] [new²] = P1

5 6 54 86 84

Zach, Beniah) (Get the Back up
from Zach db)

Notes → Actual Account customer

Transaction page

Select swaps

Use proportional reduction every 2000
Good changes or 20,000 attempted changes

REPEAT UNTIL 20,000 ATTEMPS
AND NO successful changes

④

Compatibility irsd

its true we return true \sim

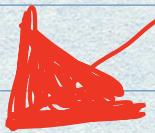
if true, then Good change \rightarrow
successful change = 0
attempted change $\neq 1$

if false, attempted change $\neq 1$
successful change $\neq 1$

CHANGES

- change swaps to bad

- swaps need to pass a T for processing
- Create organizational functions in swaps
- Be able to swap BACK



i

if $e^{(\text{changeint})}$ is less than $\text{Math.random}(0,1)$
then swap back
More

temp is adjusted 000

hotel [newroom] [ə] * hotel [newroom] [ɪ]

+ hotel [newroom] [ə] * hotel []

0 1

[newroom²] [ə]. compatibility nom [hotels[newroom²] [ɪ]] id

0 1 - 2]

4 S 6 7

AB

AC

AD

BC

BD

CD

89

15

17

79

85

7

4. Compatibility norm [E][S]