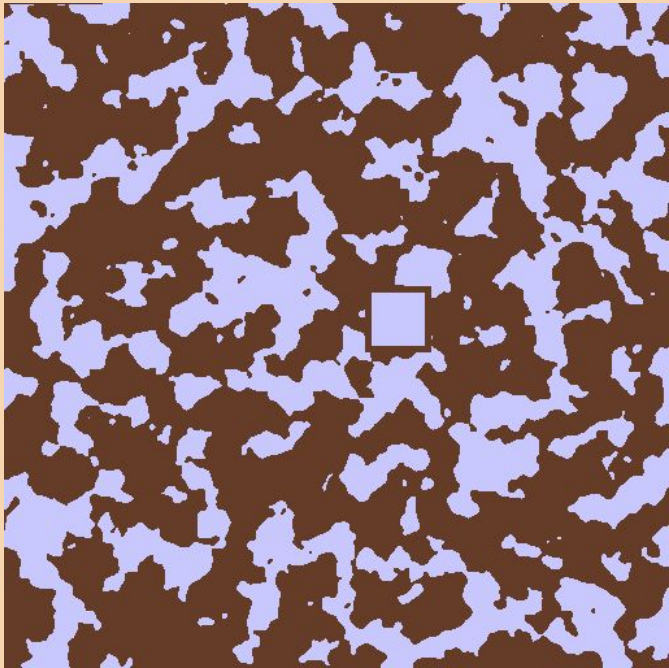
MARTIAN SQUIRREL CAVE DWELLING

Jayden Patel

ASSIGNMENT

Given is a **512x512x512** array of ones and zeros representing a section of Mars. Each **one** is a cubic meter of **dirt**, and a **zero** represents a cubic meter of **air**. The goal of the assignment is to find four locations for a **51x51x51** meter dwelling for a squirrel. The dwelling is made up of a **41x41x41** section of space with a **5** meter wall on all sides. Each cubic meter of dirt or air changed costs **one acorn**. Each dwelling created must have a max cost of **80,000** acorns. Also, **no floating dirt** is permitted (each cubic meter of dirt must be touching another). Dirt must also be **conserved** and must be dumped somewhere in the section.

Cross Section of Mars with Dwelling



STEP 1 - SURVEY THE LAND

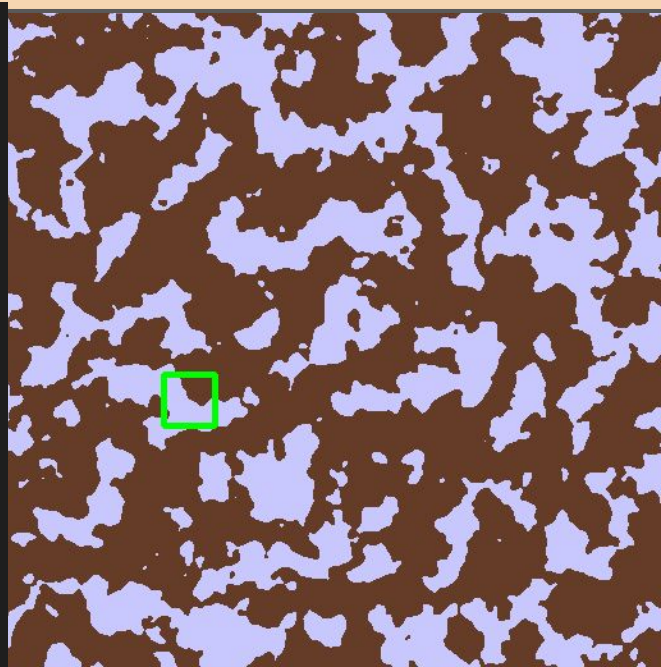
First, a dictionary of each possible x, y, and z position and the amount of dirt that must be removed from the center was created, skipping every 20 meters. **13824** positions were collected.

```
X    Y   Z  Center Dirt
(363, 83, 43) 35280
(383, 83, 43) 37716
(403, 83, 43) 40072
(423, 83, 43) 53809
```

```
def makeDwelling(cave, x,y,z):
    cave = cave.copy()
    centerDirt =
cave[x-size//2:x+size//2+1,y-size//2:y+size//2+1,z-size//2:z+siz
e//2+1].sum()

cave[x-size//2-wall:x+size//2+1+wall,y-size//2-wall:y+size//2+1+
wall,z-size//2-wall:z+size//2+1+wall] = 1

cave[x-size//2:x+size//2+1,y-size//2:y+size//2+1,z-size//2:z+siz
e//2+1] = 0
    return cave, centerDirt
```



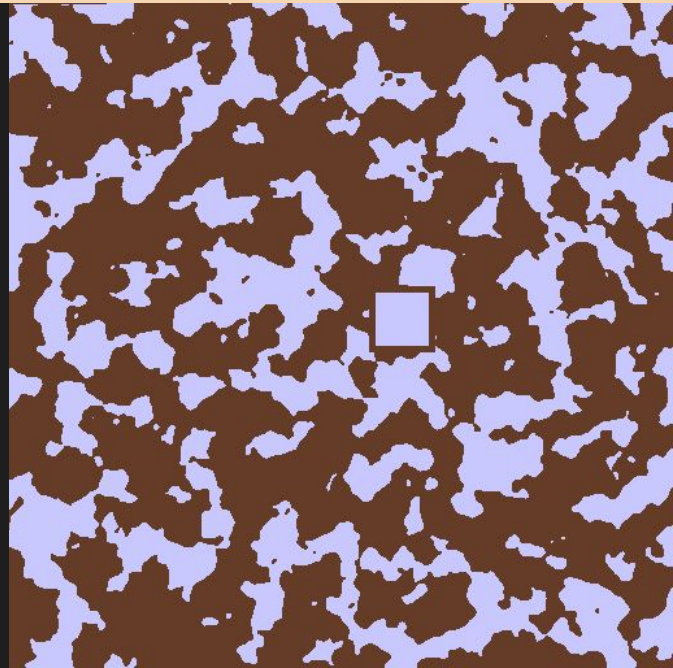
STEP 2 - CREATE THE DWELLINGS

A random result from the land survey was picked, and a dwelling was created.

```
def makeDwelling(cave, x,y,z):
    cave = cave.copy()
    centerDirt =
cave[x-size//2:x+size//2+1,y-size//2:y+size//2+1,z-size//2:z+si
ze//2+1].sum()

cave[x-size//2-wall:x+size//2+1+wall,y-size//2-wall:y+size//2+1
+wall,z-size//2-wall:z+size//2+1+wall] = 1

cave[x-size//2:x+size//2+1,y-size//2:y+size//2+1,z-size//2:z+si
ze//2+1] = 0
    return cave, centerDirt
```



STEP 3 - DUMP THE DIRT

The fillDirt function found all of the **air** within **5 meters** of the dwelling's outer wall and **filled** in the correct amount of air to offset the construction of the dwelling.

```
tempCave[x-size//2-wall:x+size//2+1+wall,y-size//2-wall:y+size//2+1+wall,z-size//2-wall:z+size//2+1+wall] = 1
slice =
tempCave[x-size//2-wall-buff:x+size//2+1+wall+buff,y-size//2-wall-buff:y+size//2+1+wall+buff,z-size//2-wall-buff:z+size/
/2+1+wall+buff]
spaces = np.argwhere(slice == 0)
spaces[:, 0] += x-size//2-wall-buff
spaces[:, 1] += y-size//2-wall-buff
spaces[:, 2] += z-size//2-wall-buff

first_dirt = 0
for i in range(len(spaces)):
    spacex, spacey, spacez = spaces[i]
    if np.sum(np.abs(original_cave[spacex-1:spacex+2, spacey-1:spacey+2, spacez-1:spacez+2] - 1)) > 0:
        first_dirt = i
        break
original_cave[spaces[first_dirt:dirt_amount,0],spaces[first_dirt:dirt_amount,1],spaces[first_dirt:dirt_amount, 2]] = 1
return original_cave
```

STEP 4 - ASSESS THE COSTS

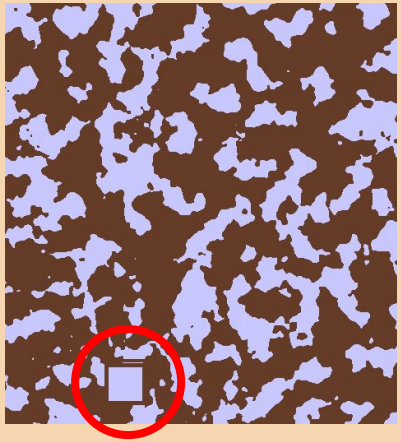
The net cost for each dwelling was calculated, and the location was **scrapped** if the cost was **above 80,000**. Otherwise, the final section was saved to a file and the next random location was processed.

```
cost = np.abs(newCave*1.0-ogCave).sum()
if cost > 80000:
    continue
```

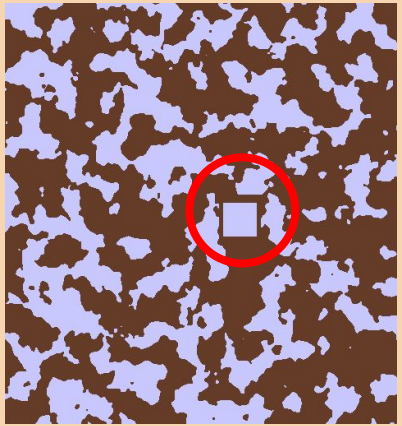
LOCATIONS

Four locations were saved from the many possible locations with their costs.

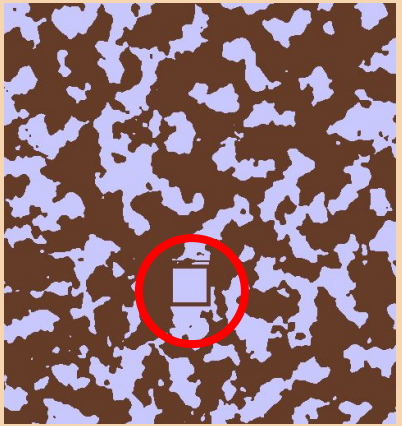
South Valley Suburb
(463, 163, 383)
56110 acorns



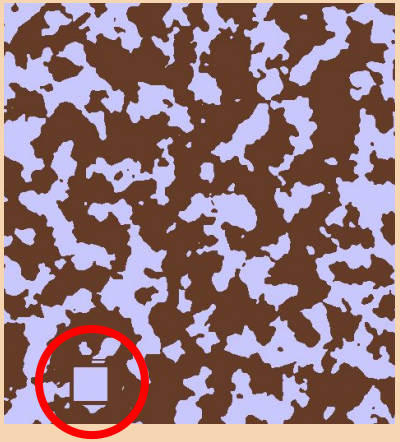
Midtown Penthouse
(263, 303, 283)
58372 acorns



Cherrywood Mansion
(343, 243, 383)
59224 acorns



Oakland Heights
(463, 123, 143)
62356 acorns



CONCLUSIONS

The cheapest dwelling found in the random processing was located at **(463, 163, 383)** and costed **56110** acorns. However, not **all** of the collected data during the survey phase was **processed**. Using faster methods such as multi-threading and improving the cost assessment algorithm would make the program more efficient.

MARTIAN SQUIRREL CAVE DWELLING

Jayden Patel