Project: Apartment Hunting

Author: Prasad Shetty P

GIVEN DATA:

List of adjoining blocks in a street with buildings (facilities available):

```
blocks= [{'gym': False, 'office': False, 'school': True, 'store': False, 'clinic': False, 'pool': True},
         {'gym': True, 'office': True, 'school': False, 'store': False, 'clinic': False, 'pool': False},
         {'gym': False, 'office': False, 'school': True, 'store': False, 'clinic': True, 'pool': True},
         {'gym': False, 'office': False, 'school': True, 'store': False, 'clinic': False, 'pool': False},
         {'gym': True, 'office': True, 'school': True, 'store': False, 'clinic': False, 'pool': False},
         {'gym': True, 'office': True, 'school': False, 'store': False, 'clinic': False, 'pool': False},
         {'gym': False, 'office': False, 'school': False, 'store': True, 'clinic': True, 'pool': False}]
```

List of required buildings (facilities to have):

```
reqs= ['gym', 'office', 'store', 'clinic', 'pool']
```

APPROACH:

- First step is to find out blocks where required facilities available.
- Listing all those indexes of blocks against each facility.
- Take each block one by one and find the distances to each location.
- Take minimum distance out of all the available options for a particular facility.
- Consider maximum distance among all the facilities for a given block and hold it in a list for all the blocks.
- Finally take out minimum in list of maximum distances for n blocks. That's the optimum distance and optimum block to be selected.

PROCEDURE:

- List out all the indexes where required buildings present and make a list of lists containing indexes of the blocks in the order of elements in reqs.

  If we take out all the indexes holds 'True' for that required building (i.e., gym, office, school l. etc.) and replacing it in place of each element of reqs. We end up getting a list of lists.

  Blocks(indexes) where required facilities available:

	gym	office	store	clinic	pool
Available at indexes:	1,4,5	1,4,5	6	2,6	0,2

Code:

avail= [[index for index, block in enumerate(blocks)if block[building]] for building in reqs]

By using list comprehension:

a) Iterate through all the buildings in list of required buildings
b) Iterate through all the individual blocks and corresponding indexes in blocks using the enumerate method (which gives out index and corresponding object)
c) Make a list of all the indexes of blocks having buildings in question and storing it in a list named 'avail'
d) Gives a list of lists:
avail= [[1, 4, 5], [1, 4, 5], [6], [2, 6], [0, 2]]

- Variable, 'avail' contains all the locations where the required buildings are available. now we must check the distances of that location from the 0th block, 1st block, 2th block…and so on till the nth block and choose lowest distance.

  For example: If 1th block is taken:

  a) Gym is available at block1, block4, block5:

  b) Distance= Subtracting [1,4,5] index values with 1: gym is at distances are 0,3,4 respectively

c) choose minimum distance to walk, i.e., 0

Similarly,

Office is available at block1, block4, block5: distances are 0,3,4 respectively, minimum distance is 0

Store is available at block6: distance is 5, minimum distance is 5

Clinic is available at block2, block6: distances are 1 and 5 respectively, minimum distance is 1

Pool is available at block0, block6: distances are 1 and 5 respectively, minimum distance is 1

If we select block 1, minimum distance to walk for each required facilities/buildings gym, office, store, clinic, pool would be 0,0,5,1,1 respectively.

- Last step is to take out maximum value from list of minimum distances to walk. That will give the maximum distance to walk for any facility if we select that block.

  For Example: for 1th block,

  a) List of minimum walking distance is: 0,0,5,1,1

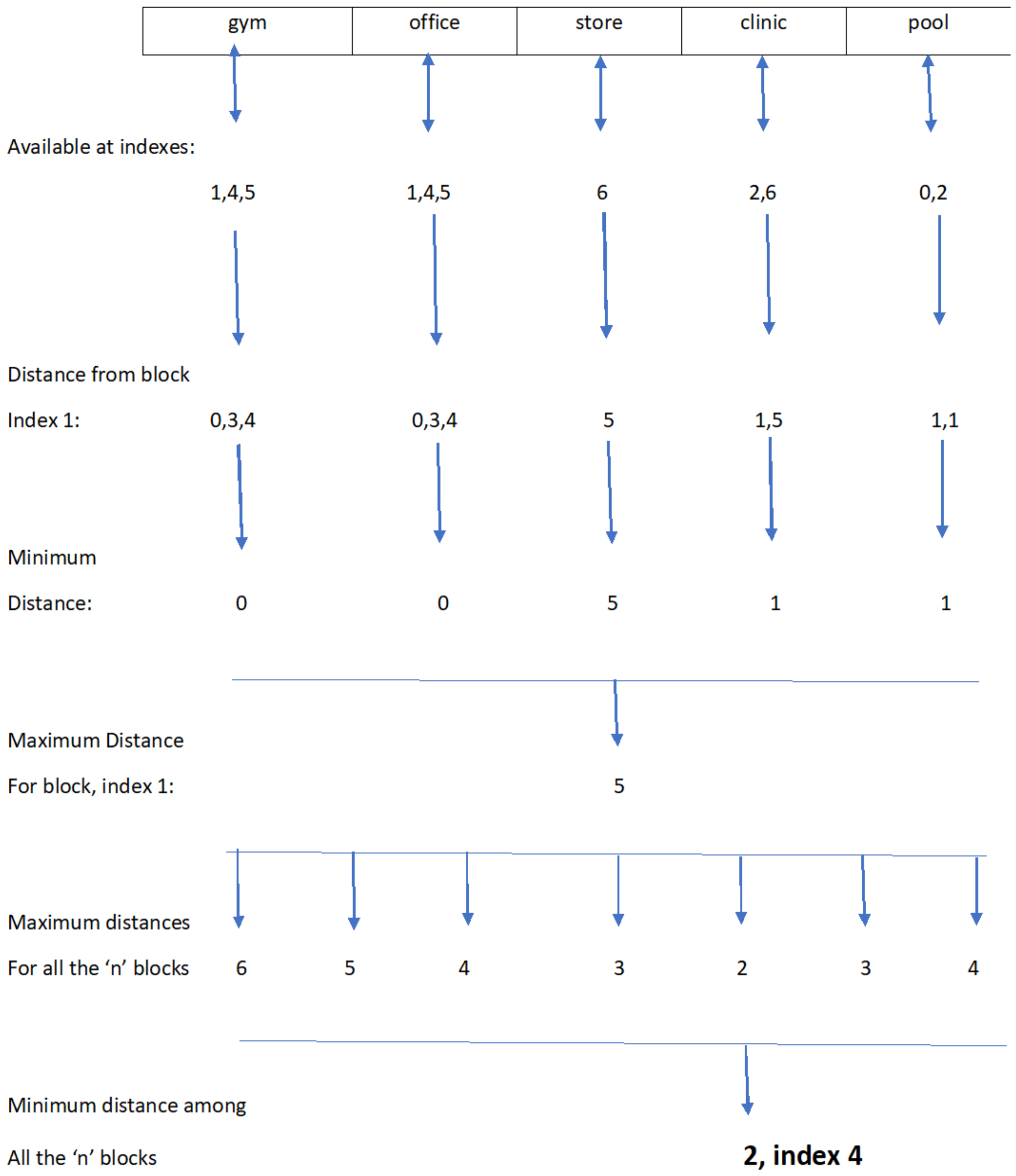
  b) Maximum distance to walk for any facility is 5.

- After computing maximum distance for all the blocks from 0th to nth block, we get the list of n values which are the maximum distances to be walked for any facility for the given block.

For 1th block, maximum distance is 5; likewise maximum distances for block indexes 0,2,3,4,5,6 are 6,4, 3,2,3,4 respectively.

that is given by the variable 'distance', which is a list

distance= [6,5,4,3,2,3,4]


gym
office
store
clinic
pool
Available at indexes:
1,4,5
1,4,5
6
2,6
0,2
Distance from block
Index 1:
0,3,4
0,3,4
5
1,5
1,1
Minimum
Distance:
0
0
5
1
1
Maximum Distance
For block, index 1:
5
Maximum distances
For all the ‘n’ blocks
6
5
4
3
2
3
4
Minimum distance among
All the ‘n’ blocks
2, index 4

➢ If we take out the minimum value from the list, ‘distance’ that would be the farthest distance to be walked to reach any of the facilities and corresponding index would be the optimum block to be chosen to minimize the distance to be walked.

Minimum of [6,5,4,3,2,3,4] is 2 and which corresponds to block index 4.

At block with index 4, the farthest we would have to walk to reach any of the required building is 2; at any other index, we would have to walk farther.

Code:

```
distance=[max(min((abs(i-ele)) for ele in item) for item in avail) for i in range(len(blocks))]
```

By using list comprehension:

a) Iterating through all the blocks from 0th to nth block.
b) Iterating through individual element in individual items in ‘avail’.
c) Taking the distance by subtracting (using abs() which ignores negative sign)the block index with indexes where particular building is available to find the distances, holding it in a list.
d) Taking the minimum value only from that list using min() function.
e) Compute for all the block indexes from 0th to nth.
f) Take out only maximum value for one block, which is maximum distance to travel for any facility. Holding that ‘n’ maximum distance values for each block index in variable called ‘distance’.

```
optimumblock=distance.index(min(distance))
 maxdistance=distance[optimumblock]
```

taking out minimum distance have to be walked from a list of maximum walking distances and corresponding index using min() and index() functions respectively.