CS431 Programming Languages Lab

Submitted by: Kartik Gupta (170101030)

Question 1

Write the algorithm (in pseudo-code) that you devised to solve the problem.

The Algorithm used was:

1. Find all the dimensions possible for all the bedrooms, bathrooms, garden, etc., and store them in lists.
2. Generate the tuples for all the possible dimensions for the various rooms.
3. Iterate over all the tuples and remove those that don't satisfy the space constraints (tuples that exceed the maximum area or when kitchen > hall/bedroom or bathroom > kitchen)
4. Choose the best tuple that maximizes the space used and doesn't exceed the maximum area.
5. Output the best tuple.

Question 2

How many functions did you use?

```
design
zipLists1
zipLists2
zipLists3
zipLists4
zipLists5
findPossibleDimensionsHelper
findPossibleDimensions
maximumArea
removeRedundant3
removeRedundant4
removeRedundant5
```

Question 3

Are all those functions Pure?

All the functions except the `design` function are pure functions.

Question 4

If not, Why?

The design function is impure because it involves an IO call. The putStrln function called is an impure function so this makes our design function impure. Function putStrln is impure because it causes output to an I/O device as a side effect.

Short Notes

Lazy Evaluation

Yes, the lazy evaluation feature of Haskell can be exploited for better performance in the solutions to the assignments, because it helps in handling lists of very big size as they are evaluated only when needed. In question 3, the computation of possible dimensions is not executed when calling the function but when it is needed. Hence, lazy evaluation saves computation time and memory.

Imperative language

Lack of side effects means that the state of the program is not changed during execution, and hence we can use parallel processing using those functions and we won't have to worry about consistency of variables. Because of this reason, there won't be many critical sections and hence we can use parallel processing. So using Haskell has a major advantage for these kinds of problems.