COMP - Dataflow Analysis: Liveness Analysis (2) (MIEIC - Compilers - 2021)

* This form will record your name, please fill your name.

1. The liveness analysis technique presented needs to be applied in a specific order of the CFG nodes to be right:

   - ◯ True
   - ◯ False

2. Visiting the CFG nodes by a forward order than a backward order of the flow when iterating during the liveness analysis possibly results in

   - ◯ more iterations
   - ◯ wrong in and out sets

3. In a pre-processing step to define the visiting order of the CFG nodes for liveness analysis, the suggestion is to use

   - ◯ breath-first search (BFS)
   - ◯ depth-first search (DFS)

4. The variables we select for liveness analysis are the local scalar variables (and possibly the function parameters) because:

- ◯ We are interested to determine the liveness of the variables for which we want to allocate registers
- ◯ They are the only variables for which we can apply the dataflow analysis algorithm;

5. In straight-line code (i.e., sequences of code without branches) what is the minimum number of iterations we need to output the liveness analysis results?

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ None of the options

6. From which liveness analysis sets (i.e., use, def, in, out) we can identify all the lifetime interferences between variables?

- ◯ in and out sets
- ◯ in sets
- ◯ out sets
- ◯ use, def, in, out sets
- ◯ use and def sets

7. Instead of applying the dataflow analysis algorithm for liveness analysis one time (e.g., doing liveness analysis for all the considered variables at the same time), can it be used to determine the liveness of a single variable each time and, assuming N variables, apply it N times (each time for determining the liveness of a different variable)?

- ◯ No
- ◯ Yes

8. Given any CFG (with N nodes) and K scalar variables, what can be the maximum number of iterations one may expect for liveness analysis (and whatever is the visiting order of the CFG nodes)?