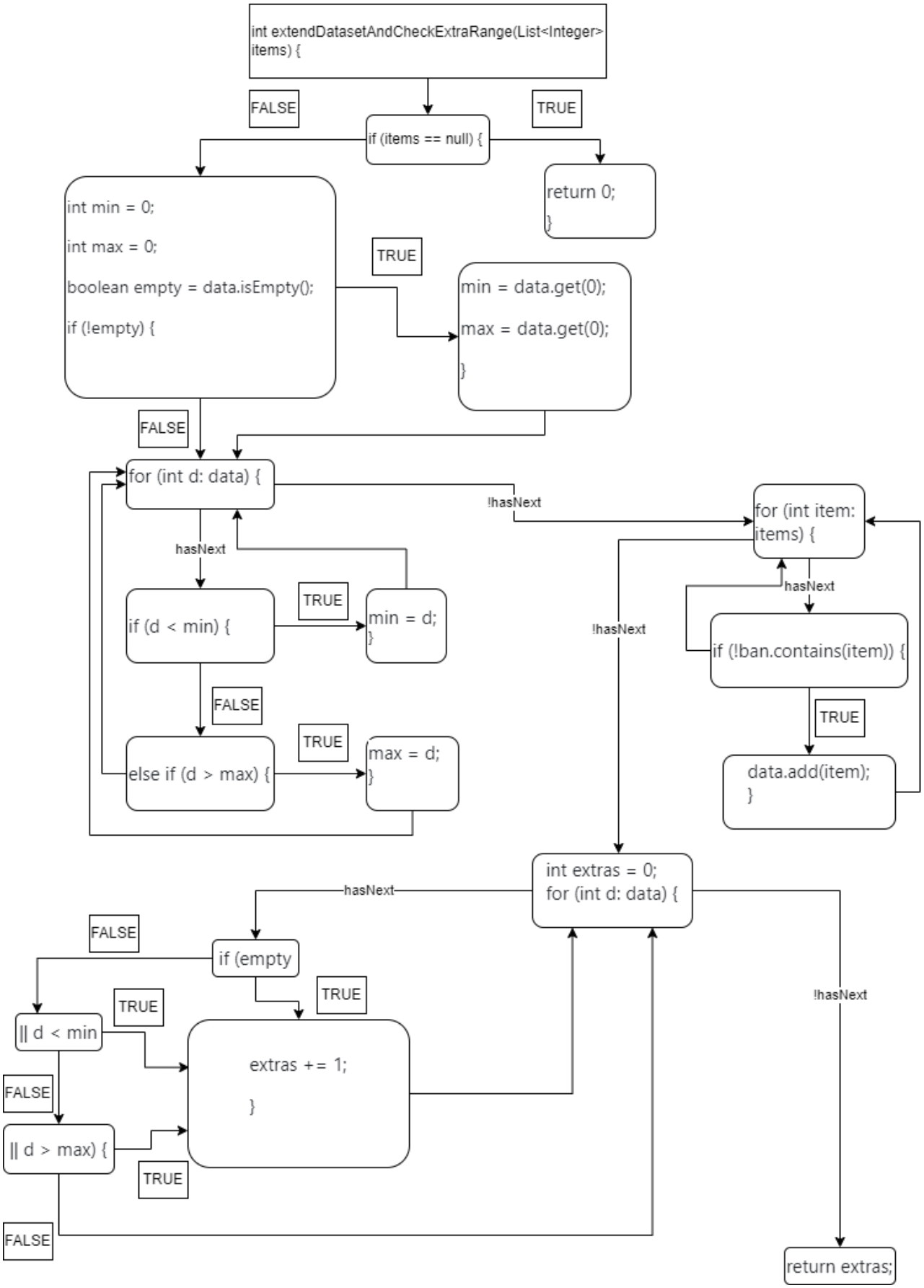
ASSIGNMENT 2

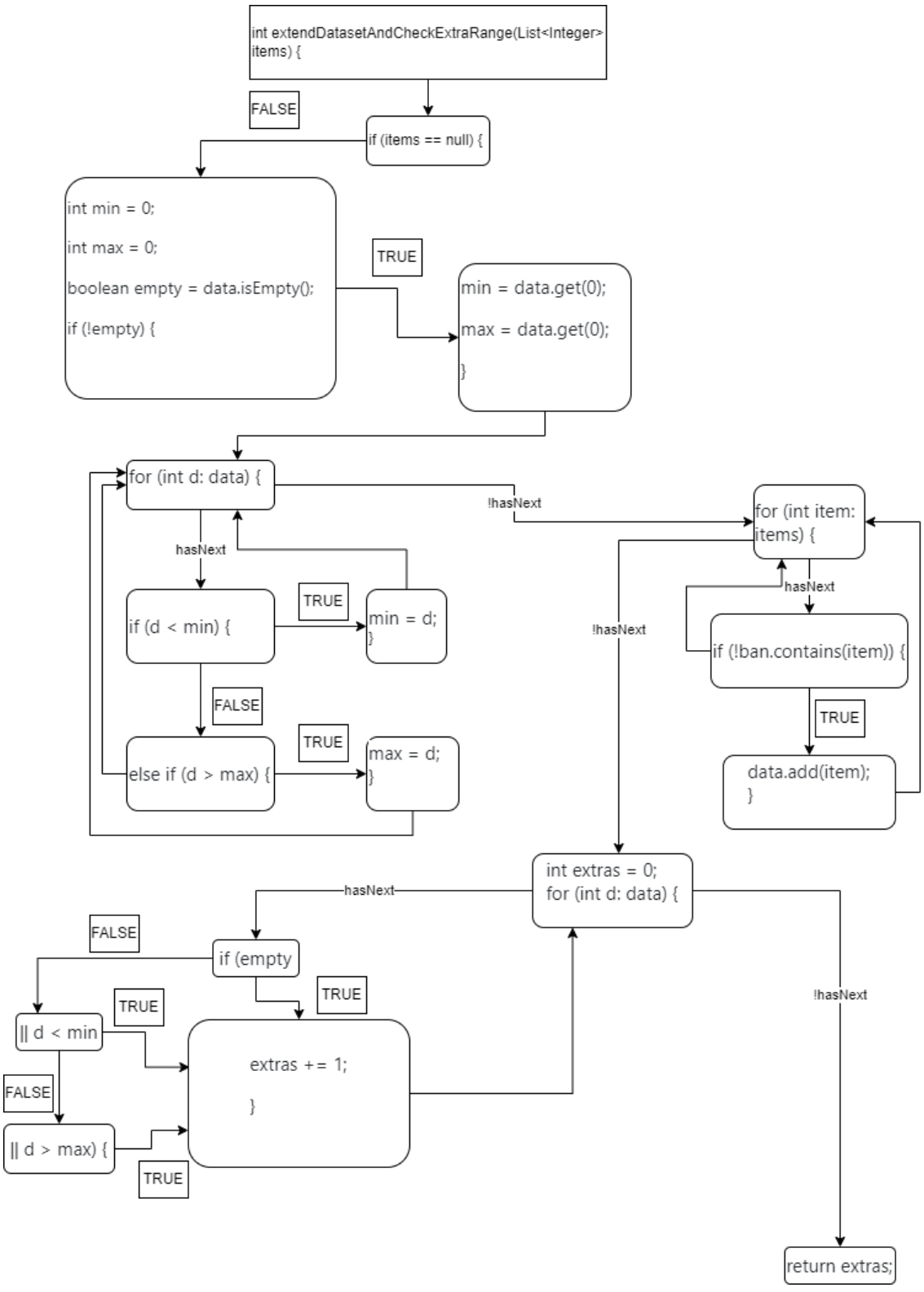
```
class Dataset {
	List<Integer> data = new ArrayList<>();
	Set<Integer> ban = new HashSet<>();
	void ban(int i) {
		ban.add(i);
	}
	int extendDatasetAndCheckExtraRange(List<Integer> items) {
		if (items == null) {
			return 0;
		}
		int min = 0;
		int max = 0;
		boolean empty = data.isEmpty();
		if (!empty) {
			min = data.get(0);
			max = data.get(0);
		}
		for (int d: data) {
			if (d < min) {
				min = d;
			} else if (d > max) {
				max = d;
			}
		}
		for (int item: items) {
			if (!ban.contains(item)) {
				data.add(item);
			}
		}
		int extras = 0;
		for (int d: data) {
			if (empty || d < min || d > max) {
				extras += 1;
			}
		}
		return extras;
	}
}
```

CFG GRAPH:


int extendDatasetAndCheckExtraRange(List<Integer> items) {
FALSE
if (items == null) {
TRUE
return 0;
}
int min = 0;
int max = 0;
boolean empty = data.isEmpty();
if (!empty) {
TRUE
min = data.get(0);
max = data.get(0);
}
FALSE
for (int d: data) {
!hasNext
for (int item: items) {
hasNext
hasNext
if (d < min) {
TRUE
min = d;
}
!hasNext
if (!ban.contains(item)) {
FALSE
TRUE
TRUE
else if (d > max) {
TRUE
max = d;
}
data.add(item);
}
int extras = 0;
for (int d: data) {
hasNext
FALSE
if (empty
TRUE
!hasNext
TRUE
|| d < min
extras += 1;
}
FALSE
|| d > max) {
TRUE
FALSE
return extras;

QUESTION 1:

In order to get a feasible path that executes the instruction ***extras+=1*** we need to have at least 1 element in ***data***. We also need to have elements in ***items*** which are not banned from ***ban*** and are also already available in ***data***. The last condition we have to meet to make this path feasible is that elements inside ***data*** satisfy the condition of the last for each loop.



QUESTION 2:

In order to get a infeasible path we need to have not met every condition inside the stament: ***if (empty || d < min || d > max) {***