Assignment 2 Rubrics (100 points max, 110 points in total)

Q1 (5 points)

- (a) Wrong or no reason -1
- (b) Wrong or no reason -1
- (c) Wrong -3、no reason -2

Q2 (10 points)

1. "Prove the following using set identities." +6
2. "Please write out the names of identities used at each step (see the lecture slides for examples)." +4

Q3 (10 points)

1. Prove it carefully.

Q4 (15 points)

1. process integrality(4p) conclusion(1p)
2. process integrality(4p) conclusion(1p) 1&2 don't use concrete examples(-4p)
3. right example(5p)

Q5 (5 points)

- reasonable explanation accounts for 5 points

Q6 (5 points)

- if you divide the cases to finite set and infinite set, then finite case for 2 points, infinite case for 3 points
- else, if just say $|A \times B| = |A| \times |B|$ get 0 points
- **P.S** Actually there is no need for classified discussion

Q7 (10 points)

- 5 sub quentions in total,each for 2 points.
- 1p for thr right judgement, 1p for the provement

Q8 (5 points)

- classification 2p
- provement for each class 3p

Q9 (10 points)

- This part has 10 points totally.
- If you tell me that, part of the origin equation can get XX, that is OK. You will get 4 points or above.
- If you simplify the equation and correct, then 10 points will give. If your answer is wrong, don't worry, you will get 5-7 points.
- If you only have the answer but not the simplification, you will get 5-7 if answer is correct.
- If you only have the answer but wrong, 0 will be given.
- IF YOU USE THE METHOD IN **'HINT'**, YOU MUST SHOW ME THE **SAME** EQUATION IN ANSWER OR GIVE A **VERY CLEAR** EXPLANATION.

Q10 (5 points)

- This part has 5 points totally.
- If your example is reasonable, you will get all the points.
- Otherwise, you will get 0 if your writing is duty, get 1-2 if your writing is beautiful.
- If you just prove that by your ways, ex. explaining, that's OK, the point judging by your explaining.
- If you don't explain for your answer, point minus 1

Q11 (5 points)

- This part has 5 points totally.
- All you need is that your reason is acceptable.
- If your direction to prove is right, you will get 3-5.
- Points will be punished if your explanation is weak.
- If you direction is wrong, give 0-2.

Q12 (10 points)

Everyone is doing so badly on this question. Please listen to the lecture and read slides carefully.

(a)

1.This question was poorly answered, so decide whether it is correct or not, as long as you carefully analyze it, give 1-2 points as appropriate.

2.If the answer is more correct in thinking, even if the process is rough, 4 points are give.

3.Some students have finished a. perfectly and hasn't proved b. or prove b. poorly, 4 points given.

The following is the original grading scale in details:

a. The set of all computer programs for a specific programing language is countable. (3 points)

The set of finite strings S over a finite alphabet A is countable (1.5 points)

Define your favorite alphabetical order for symbols in A . We show that the finite strings in S can be listed in a sequence: 1. list all the strings of length 0 in alphabetical order 2. list all the strings of length 1 in alphabetical order 3. list all the strings of length 2 in alphabetical order

The set of all Java programs is countable. (1.5 points)

Let S be the set of finite strings constructed from the finite alphabet that consists of all characters that may appear in a Java program. Define any alphabetical order for such characters. Then, as proved in the previous theorem, we can enumerate strings in S. For each enumerated string s, do the following: feed s into a Java compiler if the complier says YES (i.e., s is a syntactically correct Java program), we add s to the list, otherwise, skip it move on to the next string

(It doesn't have to be Java programs)

b.The set of all computer programs in all existing programming languages is countable. (2 points)

Then, since the number of existing programming languages is finite, one can count all computer programs in all existing programming languages as follows: for each language, fix a list that counts all programs in this particular language; in the first round, count the first programs in all those (finite many) lists; next, in the second round, count the second programs in all those lists; etc.

(b)

1.This question was poorly answered, so decide whether it is correct or not, as long as you carefully analyze it, give 1-2 points as appropriate.

2.Many students get the representative of f correctly and use Cantor’s diagonalization argument but prove wrongly, 2 points given.

The following is the original grading scale in detail:

Any function f : Z+ → {0, 1, . . . , 9} can be viewed as an infinite string a1a2 . . . an . . . where ai ∈ {0, 1, . . . , 9}, because it defines f as: f(1) = a1, f(2) = a2, . . . , f(n) = an, . . . (). (2 points)

Use Cantor’s diagonalization argument to prove that the infinite string is uncountable. (3 points)

Q13 (5 points)

1.By definition, it suffices to show that there exist constants C1,C2 and n0 such that... **Give one point for each equivalent statement**

2**.C1-> 1 point, C2 -> 1point, n0-> 1point** (Many students forgot the range of n0)

3.Some students get a constant of the ratio, it is regarded as get c1 and c2, 2 points are given.

Q14 (10 points)

(a)

1.1 point for right answer, 2 points for right process.

2.For the process, the key word is half or 2^n = ...

Time complexity: **Θ(log2 n) (or simply Θ(log n))**. The key observation is as follows: in the beginning the "distance" between i, j equals n − 1 (i.e., i = 1, j = n), then the while loop terminates only if i "meets" j (i.e., i ≥ j), and in each iteration the distance between i, j is **halved** (i.e., line 4 ~ 9).

(b)

1.1 point for right answer, 2 points for right process.

2.1 point for best-case time complexity Θ(1) and 1 point for worst-case time complexity Θ(log n).

3.For 2., if you have proved the best-case, since you can find the worst-case easily or since it is obvious, 2 points are given.

4.For 1., the answer can be different, just add such a check-then-break clause to the while loop.

In the best case, x equals the first element in the sequence, i.e., x = a⌊(n+1)/2⌋. We only need to terminate the while loop and returns m when this happens. Therefore, **one can just add such a check-then-break clause to the while loop**. For instance, we can add **"if x = am then return m**" between line 4 and line 5. It is easy to see that this new algorithm has best-case time complexity Θ(1) and worst-case time complexity Θ(log n).

(c)

2 points for right answer.

It is not hard to see that the storage of the n integers a1, a2, . . . , an dominates the space complexity, which takes Θ(n) memory.

(d)

2 points for right anwser.

Each input has the same fixed length m = ⌈log2(max{|x|, |a1|, . . . , |an|}+1)⌉+1. The input size is (n + 1)m.

Many students ignore that input includes x, which means it is n+1 integers.

Contact

If you have any questions, contact corresponding TAs:

Q1-3 12331240 潘凯歌

Q4-6 12111842 张羽乐

Q7-8 12111046 张天舒

Q9-11 12110416 刘家宝

Q12-14 12110411 伍福临

Grading: 12332414 李昊洋