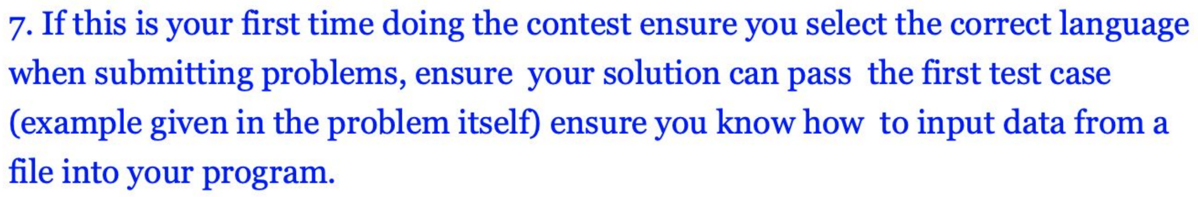
CCC Preparation

How to on CCC

CCC Email


7. If this is your first time doing the contest ensure you select the correct language when submitting problems, ensure your solution can pass the first test case (example given in the problem itself) ensure you know how to input data from a file into your program.

PS: File IO is not required for CCC. Use standard in and standard out.

Strategy for CCC Junior

- Simply solve in order.
- Most of the problems are just implementation. Read the problem, understand, and implement it with code.
- J5 may require some advanced topics. **Do it last.**

Strategy for CCC Senior

Greedy algorithm: go for easiest points first, then come back and clean up

1. Solve P1
2. Solve P2
3. Solve/partial P3
4. Partial P4
5. Partial P5
6. Solve P3/P4
7. Solve P5

Useful Data Structures

Data Structure	C++	Java	Python
Arbitrary Length Array	`vector, deque`	`ArrayList, ArrayDeque`	`list`
Arbitrary Size Set	`unordered_set`	`HashSet`	`set`
Ordered Set (Balanced Binary Search Tree)	`set, multiset`	`TreeSet`	`set`
Hashmap	`unordered_map`	`HashMap`	`dict`
Ordered Map	`map`	`TreeMap`	`dict`

Useful Methods

Java TreeMap:

```
    ceilingKey(), floorKey(), higherKey(), lowerKey()
```

Java TreeSet:

```
    ceiling(), floor(), higher(), lower()
```

C++ set:

```
    upper_bound(), lower_bound(), find()
```

pbds

Useful BBST with additional set functions: find_by_order() and order_of_key()

find_by_order(): get iterator to the kth smallest element

order_of_key(): how many elements are smaller than the key

Memorize the header lol

pbds sample code

```
using namespace __gnu_pbds;
#include <ext/pb_ds/detail/standard_policies.hpp>
typedef tree<int,null_type,less<int>,rb_tree_tag,tree_order_statistics_node_update> ordered_set;
typedef tree<int,int,less<int>,rb_tree_tag,tree_order_statistics_node_update> ordered_map;

int main(){
	ordered_set x;
	x.insert(1);
	x.insert(2);
	x.insert(5);

	x.find_by_order(1); // 2
	x.find_by_order(3); // x.end()

	x.order_of_key(2); // 1
	x.order_of_key(4); // 2
	x.order_of_key(1); // 0
}
```

Primitive types

- `int` is 32 bits: on the order of 10^9
- `long` is 64 bits: on the order of 10^{18}
- `float` is 32 bits (6 decimal places)
- `double` is 64 bits (15 decimal places)
- `long double` is 80 bits (18 decimal places)

Cheesing problems

There're many ways to cheese a problem:

Block decomposition (sqrt)

Offline queries (Mo's algorithm)

Data Structure bash

Use a different algorithm for each subtask

Binary search the answer (or other important variable)

128-bit integers / BigInteger

String hashing / double hashing

Fast / Slow

This is where you write a brute force solution and a case generator.

Run it against your fast (full) solution to find bugs and issues.

Example bash script:

```
for ((i = 0; i >= 0; i ++)) do
    echo $i
    echo $i | ./gen > in.txt
    ./slow < in.txt > slow.out
    ./main < in.txt > main.out
    diff 'slow.out' 'main.out' || break
done
```

Important Topics For Junior

- Strings
- STL data structures (TreeSet/TreeMap/set/map)
- Sorting (Java especially)
- Arrays/2-D arrays
- Loops
- If statements

Important Topics For Senior

- Sorting
- Graph Theory
  - Breadth First Search, Depth First Search, Spanning Trees, Dijkstra’s, SCCs, BCCs (bridges/articulation points)
  - LCA, Euler Tour, Centroid Decomposition (thats all i can think of, prob enough)
- Data Structures
  - Binary Indexed Tree, Segment Tree, Union Find, Hashing, SQRT Decomp
- Mathematics
  - Number Theory, Combinatorics/Probability, Modular Arithmetic, Geometry
- Dynamic Programming
  - Straightforward Dynamic Programming, may have some Game Theory mixed in or Dynamic Programming Optimizations (e.g Convex Hull Trick, Knuth’s Optimization)

Constant Optimization

- Look at https://mcpt.ca/tips
- Fast Input
- In Java: use `BufferedReader` instead of `Scanner`
- In C/C++: `#pragma GCC optimize(“Ofast”)`

Language Reference

Java: https://docs.oracle.com/javase/8/docs/api/

C++: http://cplusplus.com/

Python: https://docs.python.org/3/

Mock CCC Practice:

https://dmoj.ca/contest/nccc8s

https://dmoj.ca/contest/rccc1

ok now go practice