Data Structures

- SegTree (maybe also lazy)
- Union Find (maybe with rollback)
- Link Cut Tree (if necessary, don't really know it)

Graphs

- TopoSort
- dijkstra
- BFS/DFS
- MinCut/MaxFlow
- finding cycles
- MST
- Strongly connected components

Algorithms

Prime Sieve:

Runs in $\mathcal{O}(n)$ and `lp[i]` stores the lowest prime divisor of i. Can be used as a fast prime sieve and for factorizing lots of numbers.

```
const int N = 20;
int lp[N + 1];
void create_lp_sieve() {
  vector<int> pr;
  for (int i = 2; i <= N; i++) {
    if (lp[i] == 0) {
      lp[i] = i;
      pr.push_back(i);
    }
    for (int j = 0; i * pr[j] <= N; j++) {
      lp[i * pr[j]] = pr[j];
      if (pr[j] == lp[i])
        break;
    }
  }
}
```

- fast exponentiation
- z-function

Runs in $\mathcal{O}n$ and `z[i]` stores how long of a prefix match the string has with the substring starting at i. Can be used for lots of prefix, suffix and string matching.

```
vector<int> z_func(string s) {
  int n = s.length();
  vector<int> z(n);
  for (int i = 0; i < n; i++)
    z[i] = 0;
  int l = 0, r = 0;
  for (int i = 1; i < n; i++) {
    if (i < r)
      z[i] = min(r - i, z[i - l]);
    while (i + z[i] < n && s[z[i]] == s[i + z[i]])
```

```
      z[i]++;
    if (i + z[i] > r) {
      l = i;
      r = i + z[i];
    }
  }
  return z;
}
```

C++ Details

- important iterator functions:
  - sort
  - reverse
  - ...
- data structures and their functions:
  - set, unordered set
  - vector
  - queue, priority queue, deque
  - pair, custom structs
- hashes, custom hash functions
- min, max, pow, ... stl functions
- lambdas syntax + usecases

Math

- probability distributions, including mean and std deviation
- matrix inverse with good numerical stability
- some series limits (harmonic series, ...)
- dp optimizations
- trigonometric identities (sin, cos, tan, ...)
- gcd, lcm
- Chinese remainder theorem
- mod arithmetic + inverse

Geometry

- area of polygon
- line, point and plane intersections
- convex hull
- angle between vectors