Sqrt-heuristics, contest analysis

Gleb Evstropov
glebshp@yandex.ru

November 12, 2016

Problems are listed in order of increasing estimated difficulty.

1 Yet Another Stone Game

- Straightforward dynamic programming solution in $O(n^2)$: $dp[i][j]$ — who wins if there are i stones remaining and the last move was j.
- Key observation: values of j greater than $\sqrt{2i}$ doesn't matter.
- Compute $dp[i][j]$ for all i and j, such that the above condition holds. Total running time is $O(maxn\sqrt{maxn} + tests)$.

2 Xor and Favorite Number

- Transform the sequence to partial xors of prefixes, i.e. $b_i = a_1 \oplus a_2 \oplus \ldots \oplus a_i$. Now, for each query l and r we need to find the number of pairs (i, j), such that $l - 1 \leq i < j \leq r$ and $b_i \oplus b_j = k$.
- Keep two pointers l and r, and the number of values between them with each particular value $cnt(x)$.
- If we move any of the pointers one step left or right, we can easily recompute the number of valid pairs inside the active segment.
- The above condition is enough to apply Mo's algorithm for segment queries.

3 For Fun and Lulz

- Key observation: the minimum span in this graph has the same properties as the tree generated by random Kruskal. The expected length of the diameter of such trees is $O(\sqrt{n})$.

- Thus, to add the new edge we apply cycle's criteria — consider the path between v and u and remove the smallest edge on this path.
- Store the tree as an array of parents p_v. Only pointers along the path between v and u can change during one operation.
- The expected worktime is $O(n\sqrt{n})$.

4 Important Guests

- Find the left and right parts of the graph. Divide the set of left vertices in two parts: fat and non-fat. Vertex v is fat if $deg(v) > \sqrt{m}$.
- There are three types of cycles: containing zero fat vertices, one fat vertex and two fat vertices.
- First let's describe the way to find the cycles of length 4 with zero fat vertices. Consider each non-fat vertex u and all pairs of it's neighbours (v, w). There will be $O(m\sqrt{m})$ pairs in total. For each pair (u, w) compute the number k of its occurrences and add $k(k-1)/2$ to the answer.
- Now we want to proceed with cycles of length 4 containing at least one fat vertex. Fix some fat vertex u, mark all its neighbours in special array $used(v)$. For each non-fat vertex w traverse its list of neighbours and count the number of common neighbours with u. If this number is k we should add $k(k-1)/2$ to the answer.
- The above works in $O(m)$ for one fat vertex and there are no more than $\sqrt{m}$ of them.

5 Batman Returns

- Process all queries offline.
- Consider some border k, we want to find such answers that $a_i \leq k$ and $a_j > k$.
- Replace all values $a_i \leq k$ with 1 and all values $a_j > k$ with zeroes.
- For each position compute the first 1 after and first 0 before. Process all queries in $O(1)$ time.
- Try all borders k that look like $\sqrt{n} \cdot p$ for all integer p not exceeding $\sqrt{n}$.
- We have found all answers but such that both integers are in one bucket.
- Process each bucket separately. There are $\sqrt{n}$ elements there. For each of the segments we precompute the answer (there are $O(n)$ segments in total for one bucket). Relax the answers again in $O(q)$ time.

6 Madness and Courage

- Consider all monsters one by one. For each value x we want to store the value $health(i, x)$ — what is the minimum initial value of health the warrior with attack x should have in order to be able to beat first i monsters.
- The value $health(i, x) - health(i-1, x)$ is equal to $c_i \cdot \lceil \frac{d_i}{x} \rceil$. There are no more than $2\sqrt{d_i}$ different values of $\lceil \frac{d_i}{x} \rceil$.
- The above provides the solution in $O(n\sqrt{C}\log C)$ time.
- Instead of adding monsters one by one we can add them in buckets of size $\sqrt{n}$. Adding a bucket requires to apply $O(\sqrt{d_i}\sqrt{n})$ segment updates.
- Now, for each hero we can find out the bucket where he fails and his health right before the start of this bucket. The exact position can be found by emulating the process in the bucket for each particular hero.

7 Combinations Strike Back

- Let cnt_i be the number of elements equal to i.
- Let p_i be the number of cnt_j equal to i, i.e. the number of different elements with exactly i instances.
- The number of ways to pick exactly k elements from the set is equal to the coefficient a_k (between x^k) of the following polynomial:

  $$P(x) = (1+x)^{p_1} \cdot (1+x+x^2)^{p_2} \cdot (1+x+x^2+x^3)^{p_3} \cdot \ldots \cdot (1+x+x^2+\ldots+x^n)^{p_n}$$

- Assuming that $\sum_{i=1}^{n} p_i \cdot i = n$, the above polynomial can be computed in $O(n\log^2 n)$ time using divide and conquer technique and Fast Fourier Transform algorithm.
- However, we also have to handle multiple queries and the polynomial for each of them may differ a bit. In particular, some value p_i is decreased by one and p_{i+1} increased by one.
- There are no more than $\sqrt{2n}$ different values of p_i not equal to 0. For each of them we can compute the above polynomial, thus the total complexity will be $O(n\sqrt{n}\log^2 n)$. Of course, this is still too slow.
- One can transform the solution to $O(n\log^2 n + n\sqrt{n}$ in a truly marvelous way, which this margin is too narrow to contain.