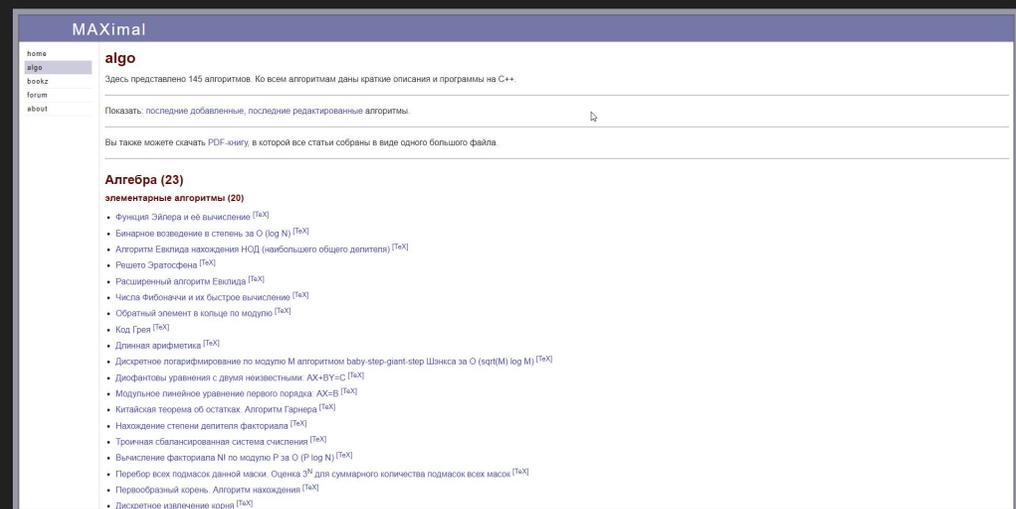
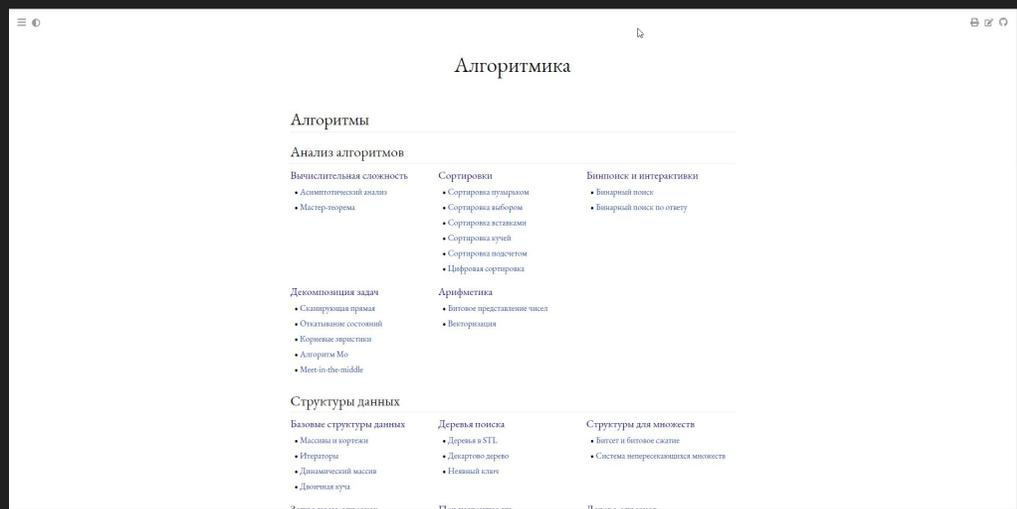


BestProger

Изучайте программирование без лишней информации

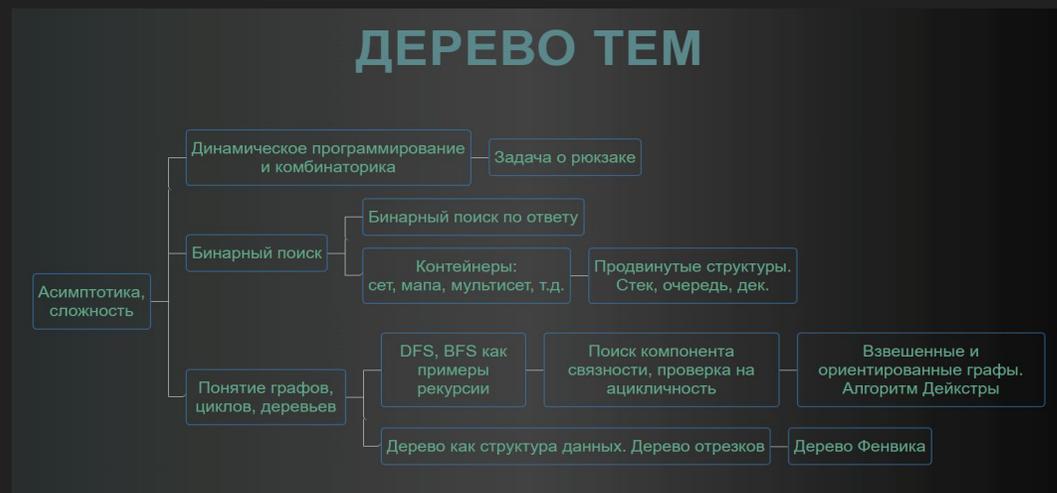
Актуальность и цель проекта

С каждым днем актуальность IT профессий растёт, а с ним и количество желающих заняться спортивным программированием на C++. Обычный пользователь заходит в интернет, с целью начать свое обучение, и натывается на сайты, **которые больше похожи на библиотеки, чем на обучающие курсы**. Информация на них хранится просто, как список тем. Это может отпугнуть пользователя, и он потеряет мотивацию. **Цель моего сайта в том, что бы любой человек, несмотря на его знания, смог начать изучение C++.**

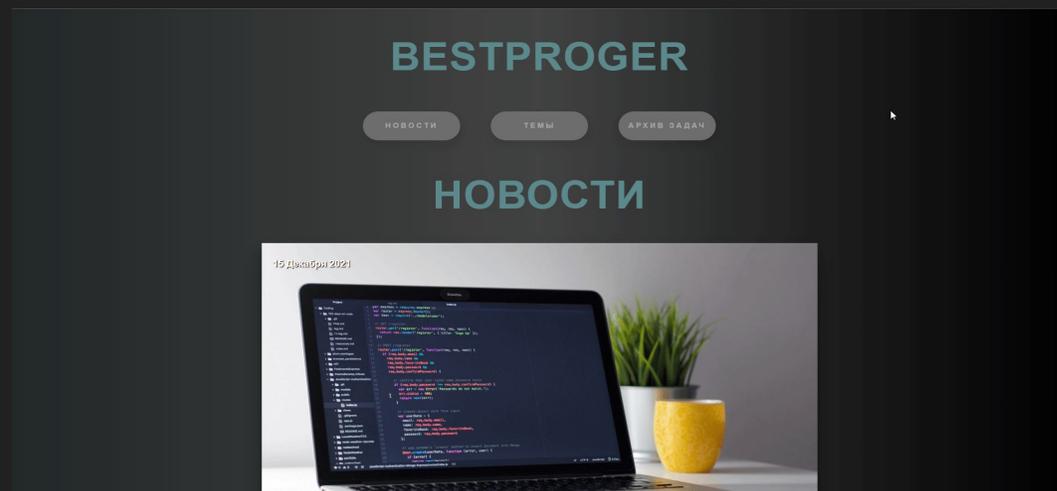


Почему именно наш сайт?

□ Удобное дерево тем, которое поможет начать изучение C++



□ Простой и приятный интерфейс, который не будет отвлекать вас от важных дел



Почему именно наш сайт?

□ Всегда под рукой есть архив задач, где можно найти практику на те темы, которые вы изучили

□ Дизайн направленный на то, что бы глаза сильно не уставали от продолжительного нахождения на сайте

включают его, и те которые не включают. Заметим, что все сочетания первой группы содержат $k-1$ элементов из первых $n-1$, а все сочетания второй группы: k из первых $n-1$. Значит,

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

Таким образом мы можем рассчитать значения C_n^k для всех $n \leq N$ и $k \leq K$ за $O(NK)$.
Реализация на C++ (значения в поле по модулю $10^9 + 7$):

```
1.#include <bits/stdc++.h>
2.using namespace std;
3.const long long MOD = 1e9 + 7;
4.long long c[1001][1001];
5.int main() {
6.    c[1][1] = 1;
7.    for (int n = 2; n <= 1000; n++) {
8.        c[n][1] = n;
9.        for (int k = 2; k <= n; k++) {
10.            c[n][k] = (c[n-1][k] + c[n-1][k-1]) % MOD;
11.        }
12.    }
13.    //Можем использовать с...
14.}
```

My Telegram

Воображаемое богатство знания — главная причина его бедности

Сложности проекта

- Так как проект нацелен на долгосрочное развитие, то в нем не должно быть сложных элементов, которые могли бы затруднить работу с проектом в будущем. Такое намерение привело к тому, что в проекте нет JS кода (Хотя его добавление планируется). При разработке сайта были использованы HTML и CSS

Усложнять просто,
упрощать сложно.

«Закон
Мейера»

- Так же нужно было придумать, как красиво отображать математические формулы. Как я уже писал выше, я не хотел затруднять проект, а значит использование костылей категорически запрещено. В итоге был найдено и подключено расширение MathJax

The logo for MathJax, featuring the word "Math" in green and "Jax" in black, with a stylized black "x" that has a small figure inside it.

Кратко про MathJax

- Подключить данное расширение очень легко, однако его использование может быть не понятно.

```
<script type="text/javascript"
  src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML">
</script>
```

Все что нужно написать, для подключения данного расширения

```
<math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
  <msubsup>
    <mi>C</mi>
    <mi>n</mi>
    <mi>k</mi>
  </msubsup>
</math>
```



$$C_n^k = \frac{n!}{k! * (n - k)!}$$

Как видно из гифки, использование MathJax представляет собой написание нужных тегов в HTML коде, что, при должном опыте, является довольно простым и понятным делом

Заключение

- На протяжении обучения в технопарке я смог получить необходимые навыки по web-дизайну, HTML, CSS, JavaScript и смог создать проект, который будет очень полезен мне и, возможно, другим людям.

Сочетанием из n по k называется любой набор из n элементов, выбранных из данных k элементов. Например, рассмотрим множество 1,2,3,4. Существует 6 сочетаний по 2 из этого множества:

- 1,2
- 1,3
- 1,4
- 2,3
- 2,4
- 3,4

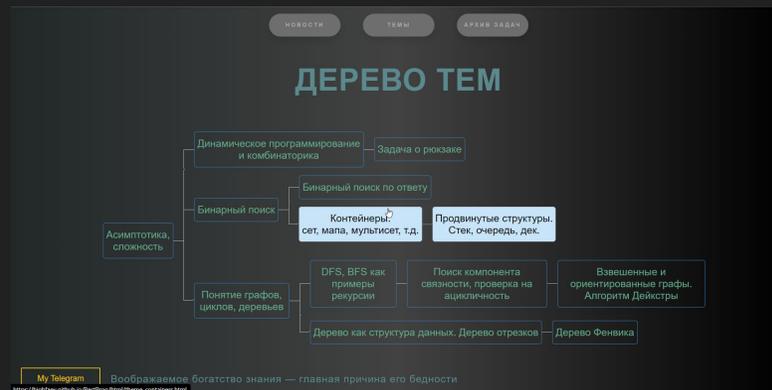
Число возможных сочетаний из n по k обозначается C_n^k и вычисляется по формуле

$$C_n^k = \frac{n!}{k! \cdot (n-k)!}$$

Число сочетаний очень часто встречается в задачах по программированию. Существуют два способа его вычисления:

Наивный способ

Используется в случаях, когда нужно посчитать небольшое количество C_n^k за $O(N)$ каждое. Реализация на C++:



```
11.     if (u != v) {
12.         dfs(u, v);
13.     }
14. }
15. }
16.
17.int main() {
18.     //Ввод графа...
19.     dfs(0);    //Начинаем обход с вершины 0.
21.}
```

С обходом деревьев связана некоторая дополнительная терминология. Вершину, с которой начинается обход называют **корнем**. Иногда в задачах корень дерева явно обозначен, а иногда его нужно выбирать самостоятельно (чаще всего эту роль выполняет первая вершина). Начав обход дерева с другой вершины (другого корня), мы получим другой порядок обхода вершин. Для обозначения такого действия есть свой термин: **подвешивание** дерева за определенную вершину.

Пару скриншотов с моего сайта

- Поставленный список задач был выполнен не полностью, однако уже на данном этапе, пользователь может изучать нужные темы. Сейчас сайт представляет из себя фигурку из пластилина, его можно легко переделать и адаптировать.

Дальнейшие планы проекта

- Добавление большего количества динамических элементов. Что бы на сайте не было так пусто.
- Доработка раздела «Архив задач». К сожалению, ему не было уделено достаточно времени.
- Добавления динамического проходящих списка олимпиад, который будет постоянно обновляться.
- Добавлений подсказок, которые будут предлагать пользователю темы для изучения, основываясь на том, что он уже прошел.
- Дать пользователю возможность писать и добавлять свои статьи через GitHub

Спасибо за внимание