

[EVOL HW1]: BFS

1) Алгоритм для обхода графа в ширину (BFS) реализован в двух вариантах:

- для графа в виде списка ребер
- для графа в виде списка смежности 2) Время работы алгоритма для графа в виде списка смежности $O(|V| + |E|)$, так как каждую вершину мы посещаем не более одного раза и смотрим не более $\sum \{ \deg(v_i) \text{ for } v \text{ in } V \} = 2 * |E|$ ребер. Для реализации в виде списка ребер мы так же посещаем все вершины не более одного раза, однако при этом просматриваем все ребра для поиска нужных - время работы $O(|V| * |E|)$. valgrind не показал особых отличий в занимаемой памяти, однако бенчмаркинг времени работы сильно отличается:
- сравнение времени работы с `opt-level = 0`:

#	task	iterations	total d
1	BFS using list of edges	189	784.2
2	BFS using adjacency list	199060	2.067

- сравнение времени работы с `opt-level = 1`:

#	task	iterations	total d
1	BFS using list of edges	242	976.0
2	BFS using adjacency list	183099	1.900

- сравнение времени работы с `opt-level = 2`:

#	task	iterations	total d
1	BFS using list of edges	249	1.095
2	BFS using adjacency list	193098	2.10

- сравнение времени работы с `opt-level = 3`:

#	task	iterations	total d
1	BFS using list of edges	257	1.157
2	BFS using adjacency list	131476	1.437

Таким образом, реализации с уровнем оптимизаций компилятора 0 и 1 получили наилучшие результаты. Большие оптимизации увеличили время работы. Кроме того, реализация на списке смежности во много раз быстрее реализации для списка ребер (что было ожидаемо).