Отчет об ИДЗ

Шавкунов Михаил

March 2, 2016

1 Сборка

Собрать программу несложно – надо написать make в консоли

2 Комментарий

Результаты тестирования находятся в папке svn. Да, я просто взял и заскринил что выводит сама программа, потому что не вижу в смысла в другом. Т.е. тем самым можно убедиться что все работает. Как видно, на маленьких тестах, Гольдберг работает не хуже(а иногда и лучше) чем Форд-Беллман. Но особо разница видна при больших тестах, когда Форд-Беллман в среднем работает в 10 раз медленне, а иногда даже в 100 раз. Это неудивительно, учитывая их асимптотику. Таким образом, я на практике в этом убедился. За N я брал примерно среднее значение равно 40. Уже на нем видна разница. Всегда можно изменить и посмотреть что выведет программа. Хотя это находится в самом генераторе...Как мне кажется это какие-то мелочи. Главное реализовано.

3 Исправление ошибок после первого review

На всякий случай оставлю здесь список замечаний по моему коду.

1. int readGraph(){//будет
   »int readGraph() { // будет (добавил три пробела)
   Исправил

2. c++11 for (int i = 0; i < top.size(); i++){ v = top[i]
   for (int v : top) {
   Исправлено.

3. for (int i = 0; i < ngraph[v].size(); i++){
   int to = ngraph[v][i];
   int edgeWeight = nweight[v][i];
   »На самом деле ты хочешь хранить массив рёбер. Так и пиши: struct Edge {...}; vector< vector<Edge> > ngraph;
   Исправлено.

4. FILE *input = fopen("input", "r"); »Имя файла лучше вынести в константу (в начале файла – #define TEST_FILE "input")
   Вынесено.

5. »По смыслу. Ты плохо генеришь тесты.
   Исправлено. Теперь два генератора. Один как несложно догадаться просто рандомный, второй генерит граф ответ на котором точно существует. С помощью второго были отловлены некоторые баги

6. int numberOfTest = 0;
   while(true){
   numberOfTest++;

   for (int numberOfTest = 0;; numberOfTest++;) {
   Исправлено.

7. По смыслу. int goal = log(N)/log(2); for 0..goal
   Исправлено.

8. for (int j = 0; j < sqrt(badVertex); j++){ int success = doIteration();
   »Тут то же самое — ты хочешь делать while (badVertex > 0) {
   Исправлено.

9. if (success == 2){ return success; } »Не понятно. Не читабельно. Видимо, ты хотел что-то типа if (result == NEGATIVE_CYCLE){
   Исправил. Теперь через enum.

10. »Твой код не должен выдавать Warning-и. Компилируй всегда с -Wall и проверяй.
    Компилировал. У меня ничего теперь не выдает.

11. srand(time(NULL)); »Это плохо, потому что тест не воспроизвести. Делай srand(count++).
    Исправлено. Теперь любой тест можно воспроизвести. Хороший однако совет!

12. rand() Не кроссплатформенно (в WIN 15-бит, в LINUX 31-бит) Вот мой аналог: inline unsigned R() { return (rand() « 15) XOR rand(); }
    Исправлено.

13. »Давай всё-таки называть исполняемый файл main =) и ключи: -O2 -Wall
    Теперь оно так.

14. const int maxn = 1e4 + 1; Раньше было все на константых массивах. Теперь программа не зависит от размера входных данных.

15. »Это относится только к генератору, стоит в нём и объявить, а не в самом верху.
    Ну теперь оно вверху, потому что стало два генератора. По-хорошему надо бы еще пытаться их вводить при запуске программы. Но это вроде мелочь. Надеюсь.

16. const int inf = 2e9; » лучше заменить на MAX_INT или MAX_INT / 2, чтобы не было меньше магических чисел.
    Изменено.