Homework 10 Solutions

1. (a) Let the graph be $G = (V, E)$ with $|V| = n$. Construct a graph $G(p)$ on V by including each $e \in E$ with probability $p = 12 \log n/(c(\epsilon/2)^2)$. By max-flow/min-cut the $s-t$ min-cut of G has value v. As in lecture, w.h.p. the $s-t$ min-cut in $G(p)$ has value at most $(1 + 2 \cdot (\epsilon/2))v = (1 + \epsilon)v$ in G. Such a cut is saturated by a $s-t$ max-flow, which can be found using the augmenting path algorithm on $G(p)$.

   Constructing $G(p)$ takes $O(m)$ time. From lecture, w.h.p. $G(p)$ has at most $(1 + \epsilon)pm$ edges and has min-cut at most $(1 + \epsilon)pv$, so augmenting paths runs for at most $(1 + \epsilon)pv$ iterations (by max-flow/min-cut). The expected number of iterations is constant (since we have w.h.p. statement), so for constant ϵ the expected running time is

$$O(m) + O(pm \cdot pv) = O(m + mv \log^2 n/c^2) = \tilde{O}(m + mv/c^2).$$

   (b) For the p above, construct $1/p$ graphs on V by independently and randomly placing each edge in one of the graphs. Note that each graph is a sampled graph, so as in lecture, w.h.p. the $s-t$ min-cut in each graph is at least $(1 - \epsilon)pv$. For each graph, run the augmenting path algorithm and return a $s-t$ max-flow. Output the union F of these flows.

   From lecture, the probability that a sampled graph does not have $s-t$ min-cut at least $(1-\epsilon)v$ is at most $O(n^{-2})$. By the union bound, the probability that one of the $1/p$ sampled graphs "fails" is at most $1/(pn^2) = O(1/n)$; therefore w.h.p. all of the graphs has a $s-t$ min-cut at least $(1-\epsilon)pv$. Since the graphs are edge disjoint, the union of their flows has value equal to their sum; therefore w.h.p. F has value $\frac{1}{p} \cdot (1 - \epsilon)pv = (1 - \epsilon)v$.

   As above, augmenting paths takes $\tilde{O}(mv/c^2)$ for each sampled graph. Since we run it on $1/p$ sampled graphs, the expected running time is $\frac{1}{p} \cdot \tilde{O}(mv/c^2) = \tilde{O}(c) \cdot \tilde{O}(mv/c^2) = \tilde{O}(mv/c)$.

   (c) The algorithm runs as follows. Construct graphs G_1, G_2 on V by placing each edge independently and randomly in G_1 or G_2. Apply the algorithm recursively on G_1 and G_2. Take the union F of the resulting flows and run augmenting paths from F to complete the max-flow (so it is always correct). As a base case we can take the case where $s-t$ max-flow is 0.

   Here, we are computing $G(p)$ (as in lecture) with $p = 1/2$, so G_1, G_2 have $m/2$ edges, $s-t$ max-flow $v/2$ and min-cut $c/2$ in expectation. Also, $\epsilon = \tilde{O}(c^{-1/2})$, so w.h.p. the $s-t$ max-flow in each graph is at least $(1-\tilde{O}(c^{-1/2}))v/2$. It follows that w.h.p. F has value $v-\tilde{O}(v/\sqrt{c})$, so augmenting paths runs for $\tilde{O}(v/\sqrt{c})$ iterations, taking $\tilde{O}(mv/\sqrt{c})$ time. The recursion is then

$$T(m, v, c) = T(m/2, v/2, c/2) + \tilde{O}(mv/\sqrt{c}),$$

   which solves to $\tilde{O}(mv/\sqrt{c})$. As in lecture, this is a probabilistic recurrence, so we need to analyze the recursion tree (as in DAUG) to show that this is in fact the running time.

2. (a) Let a, b and c be three non-collinear points. Let ℓ_{ab} and ℓ_{bc} be the perpendicular bisectors of segments $\overline{ab}$ and $\overline{bc}$, respectively. Since the points are not collinear, ℓ_{ab} and ℓ_{bc} intersect at a unique point p. This is the only point equidistant to a, b and c. Therefore p is the center of the unique circle $\mathcal{C}_p$ containing a, b, c on its boundary. Computing the perpendicular bisectors takes constant time (midpoint, slope); finding the intersection takes constant time.

   (b) **Lemma:** It is not possible to translate $O(H)$ without excluding some point of H.

   Proof. Suppose the claim is false; then there exists a direction $\mathbf{v}$ and $\epsilon > 0$ such that for all $t \in [0, \epsilon]$, $S \subset O(H)$ when $O(H)$ is translated by $t\mathbf{v}$. For $p \in H$, let $d_p(t')$ be the distance from p to $O(H)$ when $t = t'$. If $d_p(t_0) = 0$ then for some $t_p > 0$, $d_p(t_0 + t') > 0$ for $t' \in (0, t_p)$ (a point cannot stay on the boundary as the circle is translated). Therefore there exists a $\delta \in [0, \epsilon]$ such that $d_p(\delta) > 0$ for all $p \in H$. But this implies the circle at $t = \delta$ can be contracted to give a smaller circle containing H, a contradiction.

Let $\mathcal{B}(H)$ be the input points on the boundary of $O(H)$. By the setup, $|\mathcal{B}(H)| \leq 3$. If $|\mathcal{B}(H)| < 2$ then $O(H)$ can be translated in a way contradicting the lemma. If $|\mathcal{B}(H)| = 2$ then these boundary points must be endpoints of a diameter. Therefore there are $\binom{n}{2} + \binom{n}{3}$ possibilities for $B(H)$, representing $\mathcal{B}(H) = 2, 3$, respectively. Each possibility defines a unique circle: for $|\mathcal{B}(H)| = 2$, the center is at their midpoint; for three points, refer to part a. Therefore there are $O(n^3)$ circles to consider, each of which takes $O(1)$ time to define by part a. Testing if all input points are contained in a circle takes $O(n)$ time: measure the distance from each input point to the center and check if this distance is less than the radius. Therefore we can find $O(H)$ in $O(n^4)$ time.

(c) Suppose $O(H) \neq O(B(H))$. Then $O(B(H))$ excludes a point of $B(H)$ on its boundary, so there are two basis points defining a diameter of $O(B(H))$. Therefore the arc of $O(H)$ defined by the three basis points is smaller than 180 degrees. Note that $O(B(H))$ and $O(H)$ have distinct centers $c_{B(H)}, c_H$ respectively, since $c_{B(H)}$ is not equidistant to all points of $B(H)$. No other points are on the boundary of $O(H)$, so c_H can be translated toward $c_{B(H)}$ by some $\epsilon > 0$ while keeping H within $O(H)$, since translating toward $c_{B(H)}$ decreases distances from c_H to the basis points. This violates the lemma from 2b, a contradiction. Therefore $O(H) \neq O(B(H))$.

(d) Let $S' = S \cup \{p\}$. If p is not contained in $O(S)$ then by part c, $O(S) \neq O(B(S'))$. By the contrapositive of part c, $S' \subset O(B(S'))$. Suppose $B(S') \subset S$; then $O(B(S')) \leq O(S)$ (in size). Since $S \subset S' \subset O(B(S'))$, $O(S) \leq O(B(S'))$ (in size). Therefore $O(B(S'))$ and $O(S)$ have the same size. Since $O(S) \neq O(B(S'))$ the Lemma from 2b is violated, a contradiction. Therefore the assumption is incorrect, and $p \in B(S')$, so p is on the boundary of $O(S')$.

(e) Let $\mathcal{C}_H = \{B(T) : T \subseteq H\}$. For each $x \in \mathcal{C}_H$, let v_x be the number of points of H outside $O(x)$ and let the indicator i_x be 1 if x is the basis of R and 0 otherwise. Let V be the number of points of H outside of $O(R)$. Then $E[|V|] = E[\sum_{x \in \mathcal{C}_H} i_x v_x] = \sum_{x \in \mathcal{C}_H} v_x E[i_x]$. Now, $E[i_x]$ is the probability that x is the basis of R. By 2c, x is the basis of R iff all points of R are contained in $O(x)$. There are $\binom{n}{r}$ possible R. For x to be the basis, we must choose $r - |x|$ points from the $n - |x| - v_x$, since x has already been included and we cannot choose anything outside of $O(x)$. Therefore $E[i_x] = \binom{n-v_x-|x|}{r-|x|} / \binom{n}{r}$. The analysis from lecture is identical here, so

$$E[|V|] \leq \frac{|x|(n-r+1)}{r-|x|} \leq \frac{3(n-r+1)}{r-2}.$$

(f) Now define $\mathcal{C}_H = \{B(S \cap T) : T \subseteq H\}$. Let $m = |H - S|$. For each $x \in \mathcal{C}_H$, let v_x be the number of points of H outside $O(x \cap S)$ and let the indicator i_x be 1 if x is the basis of R and 0 otherwise. Let V be the number of points of H outside of $O(R \cap S)$.The bound for $E[|V|]$ is the same as above, and follows nearly identically as above, replacing n with m, except we must define $q_x = |x - S|$. Then we must choose $r - q_x$ points from the $m - q_x - v_x$ (for the same reason as above).

(g) The solution is nearly identical to SampLP. Call the algorithm SampC. Keep an active subset $S \subseteq H$, initialized to $\emptyset$. Fix an arbitrary constant c. If $n < c$ run the algorithm from 2b. Otherwise, pick a random $R \subseteq H - S$ of size at most $3\sqrt{n}$ and recursively evaluate $x \leftarrow SampC(R \cup S)$. Compute the set V of points of H outside of $O(R \cup S)$. If $|V| \leq 2\sqrt{n}$, add V to S. If $|V| = 0$, return x.

The algorithm is correct: a basis x for H is found iff no points of H are outside $O(x)$. Let $T(n)$ be the maximum expected running time when $|H| = n$. Then $T(n) \leq 6T(9\sqrt{n}) + O(n)$: since a basis element must be in V (by 2d), and by Markov's Inequality $\Pr[|V| \leq 2\sqrt{n}] \leq 1/2$, SampC is called recursively at most 6 times. Now S is initially empty, and at most $2\sqrt{n}$ points are added with each successful (bounded $|V|$) iteration, so the subproblems have size at most $6\sqrt{n}$. It takes constant time to construct $O(x)$ from x, and $O(n)$ time to determine V. The recursion follows. Repeated substitution gives $T(n) = O(n) + O(\sqrt{n}) + \ldots + O(1) = \tilde{O}(n)$.