Brief Introduction to (basic) Probabilistic Methods in CS

(Some questions might be very hard; some are only hard; a few might be not so hard. Try to solve as much as you can but don't expect to solve them all.)

1. Mimo, że temat *grafów losowych* odgrywa bardzo ważną rolę w informatyce i algorytmach, z braku czasu, temat ten był na wykładzie ledwo wspomniany.

   Dla liczby naturalnej $n \geq 2$ i liczby p, $0 \leq p \leq 1$ (gdzie p może być funkcją od n), graf losowy G (z klasy $G_{n,p}$) jest zdefiniowany w następujący sposób: G jest grafem nieskierowanym o n wierzchołkach $\{1, \dots, n\}$ i dla każdej pary wierzchołków x, y, losujemy niezależnie czy G ma krawędź (x, y) z prawdopodobieństwem p (tzn., dla każdego x, y, $\Pr[(x, y) \in E(G)] = p$).

   (i) Pokaż, że jeśli $p \leq \frac{\ln n}{20n}$ to losowy graf $G \in G_{n,p}$ jest niespójny z dużym prawdopodobieństwem.

   (ii) Pokaż, że jeśli $p \geq \frac{100 \ln n}{n}$ to losowy graf $G \in G_{n,p}$ jest spójny z dużym prawdopodobieństwem.

2. Niech $m \geq n$. Rozważmy proces wrzucania m kul losowo do n urn (tzn. każda kula jest wrzucana niezależnie z jednakowym prawdopodobieństwem trafienia do każdej urny). Oszacuj asymptotycznie maximal-load w tym procesie. Maximum load powinien być funkcją od m i n, i oszacowanie powinno zachodzić z dużym prawdopodobieństwem: conajmniej $1 - \frac{1}{n}$.

   Podaj dowód zarówno dolnej jak i górnej granicy.

   (Np., na wykładzie pokazaliśmy że maximum-load dla $m = n$ wynosi $\Theta(\log n/ \log\log n)$ z prawdopodobieństwem conajmniej $1 - \frac{1}{n}$. Twoim zadaniem jest uogólnienie tego wyniku dla dowolnego $m \geq n$.)

3. Rozważmy następujący proces wrzucania n kul do n urn, gdzie n jest liczbą parzysta. Początkowo wszystkie urny są puste. Następnie pobieramy jedną kulę za drugą i dla każdej kuli wykonujemy następujący eksperyment: wybierz (niezależnie) jedna losową liczbę ℓ z przedziału $\{1, \dots, \frac{n}{2}\}$ i jedną losową liczbę r z przedziału $\{\frac{n}{2} + 1, \dots, n\}$. Sprawdź load dla urn ℓ i r: jeśli load tych urn jest różny, to wstaw kulę do urny o mniejszym load; jeśli load jest taki sam, to wstaw kulę do urny ℓ.

   Zanalizuj maximum-load dla tego procesu. Oszacowanie asymptotyczne $\mathcal{O}(\log\log n)$ jest OK, ale lepiej by było pokazać dokładny maximum-load, z dokładnościa do stałej addytywnej. Oszacowanie powinno zachodzić z dużym prawdopodobieństwem: conajmniej $1 - \frac{1}{n}$.

   Podaj dowód górnej granicy; jeśli umiesz, sprobuj podać również dowód dla dolnej granicy.

4. Niech $n \geq 4$. Rozważmy następujacy łancuch Markowa. Mamy dany striangulowany równoboczny wielokąt wypukły o n bokach (przez wielokąt striangulowany mamy na myśli że danych jest również $n-3$ przekątnych które dzielą wielokąt na $n-2$ trójkątów). Jeden krok łańcucha Markowa zdefiniowany jest następujacą: Dla danej triangulacji, bierzemy losową przekątną $\overline{xy}$. Przekątna $\overline{xy}$ zawarta jest w dokładnie dwóch trójkątach z triangulacji, nazwijmy je $\triangle(axy)$ i $\triangle(xyb)$. Z prawdopodobieństwem $\frac{1}{2}$ usuń przekątną $\overline{xy}$ i zastąp ją przekątną $\overline{ab}$ (w ten sposób uzyskujemy nową triangulację); z prawdopodobieństwem $\frac{1}{2}$ nie rób nic.

   (i) Pokaż, że tak zdefiniowany łancuch Markowa ma rozkład stacjonarny i scharakteryzuj jaki on jest.

   (ii) Jaki jest mixing time? Postaraj się podać (udowodnić) jak najlepszą dolną granicę i jak najlepszą górną granicę.

5. Rozważmy łancuch Markowa który odpowiada błądzeniu losowemu po hiperkostce n-wymiarowej. *Hiperkostka n-wymiarowa* jest to graf of 2^n wierzcholkach taki, że każdy wierzchołek ma etykietę

będąca liczbą całkowitą pomiędzy 0 a $2^n - 1$, wszystkie wierzchołki mają rożne etykiety i dwa wierzchołki połączone są krawędzia jeśli ich etykiety w zapisie binarnym różnią się dokładnie na jednej pozycji. (I tak, np. wierzchołki 7 i 15 są polączone krawędzia, podobnie jak 0 i 1. Zauważ że stopień każdego wierzchołka jest dokładnie n.)

Pokaż (np. używając coupling lub jakiejkolwiek innej metody) jak najlepsze oszacowanie na *mixing time* dla procesu losowego błądzenia po n-wymiarowej hiperkostce, przy założeniu, że w każdym kroku losowego błądzenia z prawdopodobieństwem $\frac{1}{2}$ stoimy w miejscu, i w przeciwnym przypadku idziemy do losowego sąsiada. (Ponieważ hiperkostka jest grafem dwudzielnym, algorytm staje z pradopodobieństwem $\frac{1}{2}$ żeby uniknąć okresowości.)

6. Niech $C = (V, E)$ będzie cyklem o n wierzchołkach z identycznymi wagami na krawędziach (można więc założyć, że wszystkie wagi są równe 1). Dla takiego grafu zdefiniujmy przestrzeń metryczną (V, D_C) tak, że odległość pomiędzy parą wierzchołków v i u (tzn. $D_C(v, u)$) jest równa długości najkrótszej ścieżki w C pomiędzy v i u.

   Przypomnijmy, że dla dowolnego drzewa z dodatnimi wagami T na zbiorze wierzchołków U, *tree-metric* (U, D_T) zdefiniowane przez T jest przestrzenią metryczną nad U taką, że dla każdej pary wierzchołków $x, y \in U$, $D_T(x, y)$ jest długoscią najkrotszej ścieżki w T pomiędzy wierzchołkami x i y.

   (i) Pokaż, że dla każdego drzewa z dodatnimi wagami T na zbiorze wierzchołków U, gdzie $V \subseteq U$, *tree-metric* (U, D_T) zdefiniowane przez T musi mieć *distortion* conajmniej $\Omega(n)$ względem metryki (V, D_C). (Tzn., że dla każdego takiego drzewa T, jeśli $D_C(x, y) \le D_T(x, y)$ dla każdych $x, y \in V$, to wtedy istnieje para wierzchołków $p, q \in V$ dla której dla pewnej stałej $c > 0$ zachodzi: $D_T(p, q) > c \cdot n \cdot D_C(p, q)$.)

   (ii) Pokaż, ze można skonstrułować $N = n^{O(1)}$ tree-metrics $(U, D_{T_1}), (U, D_{T_2}), \dots, (U, D_{T_N})$ zdefiniowane przez drzewa $T_1, T_2, \dots, T_N$ na zbiorze wierzchołków U, $V \subseteq U$, tak, że:

   - dla każdego T_i i dla każdej pary $p, q \in V$ zachodzi $D_C(p, q) \le D_{T_i}(p, q)$, oraz
   - dla każdej pary wierzchołków $p, q \in V$, *oczekiwane distortion* względem losowego wyboru tree-metric T_i jest co najwyżej 2 (tzn., $\frac{1}{N}\sum_{i=1}^{N} D_{T_i}(p, q) \le 2 \cdot D_C(p, q)$).

   (iii) Czy powyższe zadanie da się również rozwiązać jeśli wagi na cyklu C są dowolne (tzn., nie muszą być takie same)?

7. Rozpatrzmy formułę SAT ψ z n klauzulami w której każda klauzula ma dokładnie ℓ literałów.

   (a) Podaj (randomized) algorytm który w wielomianowym czasie znajduje wartościowanie zmiennych spełniające conajmniej $n \cdot (1 - 2^{-\ell})$ klauzul.

   (b) *Zderandomizuj* powyższy algorytm, tak, żeby uzyskac deterministyczny algorytm który dla danej formuły SAT ψ z n klauzulami w której każda klauzula ma dokładnie ℓ literałów, znajdował w czasie wielomianowym wartościowanie zmiennych spełniające conajmniej $n \cdot (1 - 2^{-\ell})$ klauzul.

8. Niech λ będzie nieujemną liczbą, D będzie dowolną liczbą całkowitą i niech $c \ge 1$ będzie dowolną stałą. Rozważmy dowolny ciąg zmiennych losowych $X_0, X_1, \dots$ (w szczególności, zmienne losowe $X_0, X_1, \dots$ mogą być *zależne*) dla którego następujące warunki są spełnione:

   (a) $0 \le X_0 \le D$,

   (b) dla każdego t zachodzi $|X_{t+1} - X_t| \le c$,

   (c) dla każdego t, jeśli $X_t > 0$ to $E[X_{t+1} - X_t | X_0, X_1, \dots, X_t] \le -\lambda$,

   (d) dla każdego t, jeśli $X_t \le 0$ to $X_{t+1} = 0$.

   (i) Pokaż, że jeśli $\lambda > 0$ to dla pewnego $T = \Theta(D/\lambda + 1/\lambda^2)$ zachodzi $\Pr[X_T > 0] \le 0.1$.

   (ii) Czy składnik $1/\lambda^2$ jest konieczny?

   (iii) Jak należałoby zmodyfikować tezę, żeby zachodziła ona również dla $\lambda = 0$?