Nierówności Chernoffa

Zadanie 1. Niech X będzie zmienną losową mówiącą ile razy spośród n niezależnych rzutów kostką, wypadła 6-tka. Ogranicz $\mathbf{P}(X \geqslant n/4)$ za pomocą nierówności Markova, Czebyszewa oraz Chernoffa. Podaj wartości numeryczne dla $n = 1000$.

Zadanie 2. Wyznacz prawdopodobieństwo uzyskania 55 orłów w 100 rzutach symetryczną monetą. Otrzymany wynik porównaj z nierównością Chernoffa. Przeprowadź podobne rozumowanie dla 550 orłów w 1000 rzutach.

Zadanie 3. Kasyno testuje nową maszynę. Aby zagrać gracz płaci \$1, zaś maszyna z prawdopodobieństwem $\frac{4}{25}$ wypłaca \$3, z prawdopodobieństwem $\frac{1}{200}$ wypłaca \$100 a w pozostałych przypadkach nic nie wypłaca. Ku zaskoczeniu pracowników kasyna w ciagu pierwszego miliona gier maszyna przegrała $\$10,000$. Oszacuj przy pomocy nierówności Chernoffa prawdopodobieństwo zajścia takiego zdarzenia. Numerycznie wyznacz najlepsze możliwe ograniczenie (znajdź odpowiednie t).

Zadanie 4. Niech $X = X_1 + \ldots + X_n$, gdzie X_i są niezależnymi zmiennymi losowymi z rozkładu jednostajnego na zbiorze $\{0, 1, 2\}$. Wyprowadź nierówność Chernoffa dla zmiennej X, tj. ogranicz od góry następujące prawdopodobieństwa: $\mathbf{P}(X \geqslant (1+\delta)\, n)$ oraz $\mathbf{P}(x \leqslant (1-\delta)\, n)$, dla $0 < \delta < 1$.

Zadanie 5. Rzucamy n kul w sposób jednostajny i niezależny do n urn. Niech X_i będzie indykatorem na zdarzenie - i-ta urna jest pusta. Niech $Y_1, \ldots, Y_n$ będą niezależnymi zmiennymi wskaźnikowymi z prawdopodobieństwem sukcesu $p = \left(1 - \frac{1}{n}\right)^n$. Niech $X = \sum_{i=1}^{n} X_i$ oraz $Y = \sum_{i=1}^{n} Y_i$.

(i) Pokaż, że dla każdego $k \geqslant 1$, zachodzi $\mathbf{E}[X_1 X_2 \ldots X_k] \leqslant \mathbf{E}[Y_1 Y_2 \ldots Y_k]$.
(ii) Pokaż, że dla każdego $t \geqslant 0$, zachodzi $\mathbf{E}\left[e^{tX}\right] \leqslant \mathbf{E}\left[e^{tY}\right]$.
(iii) Wyznacz ograniczenie Chernoffa dla $\mathbf{P}(X \geqslant (1+\delta)\, \mathbf{E}[X])$.

Zadanie 6 (Randomizowany QuickSort)**.** Spójrzmy na QuickSorta z innej strony. Każdy element x należący do sortowanej tablicy jest wybierany na *pivota* dokładnie raz podczas działania algorytmu. Zaczynamy od nieposortowanej n-elementowej tablicy. Losujemy element x, który będzie pełnił rolę pivota podziału. Niech ten element x stanie się korzeniem drzewa binarnego budowanego 'na boku'. Jego lewym poddrzewem będzie drzewo binarne wytworzone rekurencyjnie z elementów mniejszych od x w podziale, zaś prawym poddrzewem będzie analogiczne drzewo powstałe z elementów większych od x w podziale. Tak więc konkretnemu wykonaniu algorymu QuickSort odpowiada dobrze określone drzewo binarne kodujące historię wyborów elementów na pivoty. Mówimy, że element x jest *zły* jeśli w momencie wyboru na pivota dzieli tablicę na dwie części z których jedna jest conajmniej dwa razy większa od drugiej. W języku drzewa przekłada się to na fakt, że jedno z poddrzew jest conajmniej 2 razy większe na liczność od drugiego. Dopełniając opisu x jest *dobry* jeśli nie jest *zły*.

(i) Pokaż, że na konkretnej ścieżce z korzenia do liścia drzewa, liczba *dobrych* węzłów jest niewiększa niż $c \log_2 n$ dla pewnego $c > 0$.

(ii) Pokaż, że z prawdopodobieństwem co najmniej $1 - \frac{1}{n^2}$ liczba węzłów na konkretnej ścieżce od korzenia do liścia jest niewiększa niż $c' \log_2 n$ dla pewnego $c' > 0$.

(iii) Pokaż, że z prawdopodobieństwem co najmniej $1 - \frac{1}{n}$ liczba węzłów na najdłuższej ścieżce od korzenia do liścia jest niewiększa niż $c' \log_2 n$.

(iv) Uzasadnij, że z prawdopodobieństwem co najmniej $1 - \frac{1}{n}$ czas działania QuickSorta to $\mathcal{O}(n \log n)$.

Hmm...