

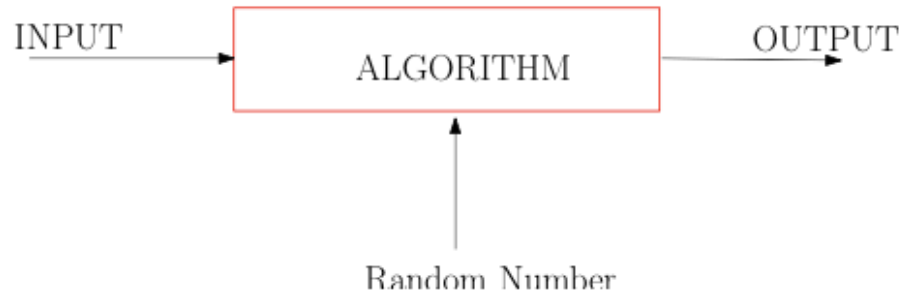
ANALISIS PROBABILISTIK DAN RANDOMIZED ALGORITHMS

KULIAH ANALISIS ALGORITMA DAN KOMPLEKSITAS

RANDOMIZED ALGORITHMS VS ANALISIS PROBABILISTIK ALGORITMA

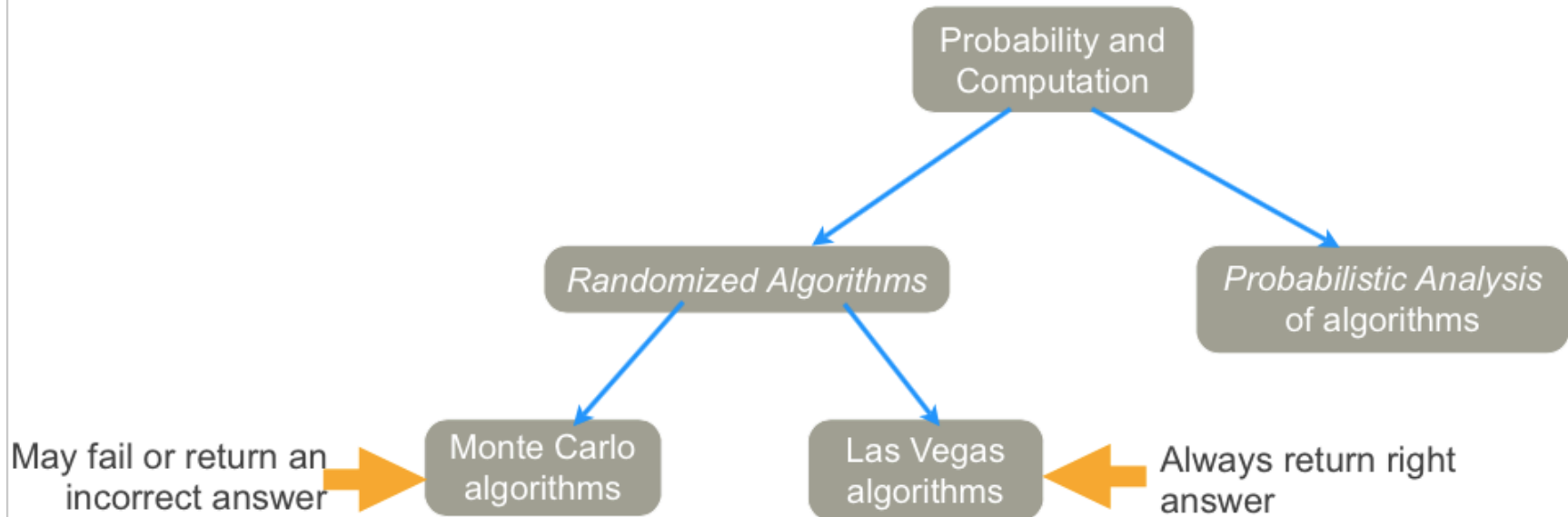
- **Randomized algorithms**
 - Algoritma yang melakukan keputusan random selama eksekusinya
 - Contoh: Quicksort dengan random pivot
- **Probabilistic analysis of algorithms (analisis probabilistik algoritma)**
 - Menggunakan teori probabilitas untuk menganalisis *behavior dari algoritma (randomized atau deterministik)*
 - Contoh: menentukan probabilitas tumbukan (collision) pada hash function

RANDOMIZED ALGORITHM



- **Sebagai tambahan pada input, algoritma menggunakan suatu sumber untuk bilangan pseudo random.**
 - Pada saat dieksekusi, dilakukan pemilihan secara random sesuai bilangan random yang dibangkitkan.
- **Behavior (output) bisa bervariasi meskipun algoritma bekerja pada input yang sama.**
- **Keuntungan:**
 - Algoritma biasanya sederhana dan mudah diimplementasikan
 - Algoritma berjalan cepat dengan probabilitas tinggi, atau/dan
 - Algoritma menghasilkan output yang optimum dengan probabilitas tinggi.

TYPE RANDOMIZED ALGORITHMS



- **Monte Carlo: randomized algorithm yang selalu dapat diselesaikan dalam waktu polinomial, tetapi mungkin mendapatkan hasil yang eror**
- Solovay–Strassen primality test: digunakan untuk mengetes apakah suatu bilangan merupakan bilangan prima.
 - Selalu menghasilkan true untuk bilangan prima.
 - Untuk bilangan komposit, menghasilkan jawaban false dengan probabilitas minimal $\frac{1}{2}$ dan true dengan probabilitas maksimal $\frac{1}{2}$.
 - Artinya, jawaban selalu benar, tetapi jawaban benar belum tentu \rightarrow $\frac{1}{2}$ -correct false-biased algorithm.
- **Las Vegas: randomized algorithm yang selalu mendapatkan hasil yang benar, tetapi running time-nya bervariasi.**
- Randomized QUICKSORT

MENGAPA PERLU RANDOMIZED ALGORITHMS?

- Banyak NP-hard problems yang mungkin mudah diselesaikan dengan “typical” inputs
- Salah satu pendekatan: menggunakan heuristics untuk meng-handle input yang tidak biasa
- Pendekatan lain: menggunakan randomization (input atau algoritma) untuk mereduksi kemungkinan terjadinya worst-case behavior

ANALISIS PROBABILISTIK

- **Adalah penggunaan probabilitas dalam menganalisis masalah.**
 - Probabilitas dapat digunakan untuk menganalisis running time algoritma.
- **Syarat: diketahuinya, atau diasumsikannya, distribusi dari input.**
- **Langkah-langkah:**
 - Menentukan/mengasumsikan distribusi probabilitas.
 - Menganalisis item yang bersangkutan berdasarkan distribusi probabilitasnya → menghitung rata-rata running time untuk semua input yang mungkin (average-case running time).
- **Isu:**
 - Performance dari input tertentu ada kemungkinan jauh lebih jelek.
 - Jika distribusinya salah, analisis dapat mengakibatkan gambaran yang tidak benar.

MENGAPA PROBABILISTIC ANALYSIS OF ALGORITHMS?

- **Jika algoritma melakukan keputusan random, maka performance tidak deterministik.**
- **Behavior dari algoritma deterministik juga bisa bervariasi antar input**
- **Probabilistic analysis juga memungkinkan estimasi batasan (bounds) pada behavior**

HIRING PROBLEM

- **Anda memerlukan pegawai baru.**
- **Anda memanfaatkan jasa agen penyedia tenaga kerja.**
- **Agan tersebut mengirimkan 1 kandidat setiap hari.**
- **Anda mewawancarai kandidat tersebut, kemudian memutuskan apakah akan mempekerjakannya atau tidak.**
- **Anda harus membayar fee ke agen setiap kali mewawancarai kandidat.**
- **Tetapi biaya lebih besar harus dikeluarkan untuk mempekerjakan kandidat.**
- **Anda berkomitmen untuk selalu mempekerjakan pegawai yang paling baik.**
- **Jadi diputuskan bahwa setelah mewawancarai kandidat, jika kandidat tersebut lebih baik daripada pekerja sekarang, maka Anda akan mengganti pekerja sekarang dengan kandidat itu.**

ALGORITMA HIRING PROBLEM

- **Input**

- Sederet n kandidat untuk suatu posisi tertentu.
- Masing-masing memiliki kualitas yang berbeda yang dapat ditentukan pada saat wawancara

- **Algoritma: Hire(n)**

- $best = 0$
- for $i = 1$ to n
 - Wawancara kandidat i .
 - Jika kandidat i lebih baik daripada $best$ maka
 - hire(i)
 - $best = i$

- **Cost: biaya yang harus dikeluarkan dari mewawancarai dan mempekerjakan.**
 - Misal biaya untuk mewawancarai setiap kandidat adalah c_i , biaya untuk mempekerjakan kandidat adalah c_h .
 - Misalkan jumlah kandidat yang dipekerjakan: n .
 - Maka total biaya yang dikeluarkan: $O(nc_i + mc_h)$.
- **Worst-case:**
 - Setiap kandidat dipekerjakan \rightarrow kapan terjadi?
 - Best case?

ASUMSI DISTRIBUSI INPUT PADA HIRING PROBLEM

- **Kandidat datang secara random.**
- **Kita dapat menentukan kandidat yang lebih baik di antara kandidat i dan kandidat $i+1$.**
 - Kandidat dapat dirangking.
 - Terdapat $n!$ kemungkinan rangking \rightarrow rangking membentuk uniform random permutation, jadi masing-masing permutasi $n!$ yang mungkin, dapat muncul dengan probabilitas yang sama.

RANDOMIZED ALGORITHMS UNTUK HIRING PROBLEM

- **Kita tidak mengetahui apakah agen mengirimkan kandidat untuk diwawancarai secara random.**
- **Agar kita dapat lebih mengontrolnya:**
 - Agen mengirimkan daftar n kandidatnya.
 - Setiap hari kita memilih secara random kandidat mana yang akan diwawancarai.
- **Algoritma juga dikatakan randomized jika inputnya dihasilkan oleh random-number generator (pembangkit bilangan random).**
 - $\text{RANDOM}(a, b)$ menghasilkan bilangan random antara a dan b dengan probabilitas kemunculan masing-masing bilangan sama.
- **Expected running time: running time dari randomized algorithm.**

VARIABEL RANDOM INDIKATOR

- **Variabel random indikator menyediakan metode untuk mengkonversi probabilitas dan ekspektasi.**
- **Misalkan diberikan sample space S dan kejadian A . variabel random indikator $I\{A\}$ didefinisikan sebagai:**
$$I\{A\} = \begin{cases} 1 & \text{jika } A \text{ terjadi,} \\ 0 & \text{jika } A \text{ tidak terjadi.} \end{cases}$$
- **Contoh: melempar koin.**
 - $S = \{H, T\}$
 - $\Pr\{H\} = \Pr\{T\} = \frac{1}{2}$
 - X_H = variabel random indikator yang berkaitan dengan munculnya H.
 - $X_H = I\{H\} = \begin{cases} 1 & \text{jika } H \text{ terjadi,} \\ 0 & \text{jika } T \text{ terjadi.} \end{cases}$
 - $E[X_H]$ = ekspektasi jumlah munculnya head dalam 1 kali lemparan koin.
 - $E[X_H] = E[I\{H\}] = 1 \Pr\{H\} + 0 \Pr\{T\} = \frac{1}{2}.$

- **Lemma: diberikan sample space S dan kejadian A pada S . Misalkan $X_A = I\{A\}$, maka $E\{X_A\} = \Pr\{A\}$.**
- **Misalkan X_i adalah variabel random indikator yang berkaitan dengan kejadian lemparan koin ke- i dengan hasil head.**
 - Jika X adalah variabel random yang menunjukkan jumlah total munculnya head dalam n lemparan koin:

$$X = \sum_{i=1 \text{ to } n} X_i.$$
 - Maka $E[X] = E[\sum_{i=1 \text{ to } n} X_i]$
 - Untuk n pelemparan koin dengan munculnya head:
 - $E[X] = E[\sum_{i=1 \text{ to } n} X_i] = \sum_i E[X_i] = \sum_i \frac{1}{2} = n/2$ (linearity of expectation)

MENGANALISIS HIRING PROBLEM

- **Kita akan menghitung ekspektasi jumlah pegawai baru yang kita pekerjakan.**
 - X_i : variabel random indikator yang berkaitan dengan kandidat ke- i dipekerjakan.
 - $X_i = I\{\text{kandidat } i \text{ dipekerjakan}\}$
 $= 1$, jika kandidat i dipekerjakan
 0 , jika kandidat i tidak dipekerjakan.
 $X = X_1 + X_2 + \dots + X_n$.
 - $E[X_i] = \Pr\{\text{kandidat } i \text{ dipekerjakan}\} = 1/i$ (karena kandidat datang secara random, probabilitas bahwa kandidat i paling baik adalah $1/i$)
 - $E[X] = E[\sum_{i=1 \text{ to } n} X_i] = \sum_{i=1 \text{ to } n} E[X_i] = \sum_{i=1 \text{ to } n} 1/i$
 $= \ln n + O(1)$ (deret harmonik)
- **Lemma: jika kandidat datang dengan urutan random, maka average case biaya total untuk mempekerjakan pada algoritma Hire adalah $O(c_h \ln n)$.**
 - Merupakan improvement dari worst case: $O(c_h n)$.

ALGORITMA RANDOMIZED-HIRE

- **Masalah hiring:**
 - Diasumsikan bahwa kandidat datang dengan urutan random.
 - Algoritma deterministik.
 - Jumlah mempekerjakan pegawai tergantung pada input.
 - Misalkan algoritma melakukan randomisasi terhadap permutasi dari kandidat, dan kemudian menentukan kandidat terbaik.
 - Algoritma randomized-hire:
lakukan permutasi secara random terhadap daftar kandidat
best = 0
for i = 1 to n
 - Wawancara kandidat i.
 - Jika kandidat i lebih baik daripada best maka
 - hire(i)
 - best = k
 - Maka algoritma tersebut adalah randomized algorithm.
- **Lemma: biaya hiring problem dengan algoritma randomized-hire adalah $O(c_h \ln n)$.**

CONTOH: PERMUTASI ARRAY SECARA RANDOM

- **Misalkan diberikan array yang memuat elemen 1 sampai n . Tujuan kita adalah menghasilkan permutasi dari array tersebut secara random.**

- **Metode : melakukan permutasi terhadap array.**
 - Pada iterasi ke- i , dipilih elemen $A[i]$ secara random di antara elemen ke $A[i]$ dan $A[n]$.
 - Algoritma:
 - Permutasi-langsung(A)
 - $n = A.length$
 - For $i = 1$ to n swap $A[i]$ dengan $A[random(i,n)]$
 - Kompleksitas: $O(n)$

PENGGUNAAN RANDOMIZED ALGORITHM

- **Polling**
- **Algoritma genetika**

CATATAN

- **Randomized algorithms bukan merupakan analisis probabilistik terhadap expected running time.**
- Randomized algorithm mengusahakan agar algoritma deterministik berubah menjadi randomized algorithm.
- Sedangkan analisis probabilitas:
 - Mengasumsikan bahwa input berasal dari distribusi probabilitas.
 - Tujuannya adalah untuk menghitung expected running time dari algoritma.

KUIS

- Berikan contoh lain dari randomized algorithm yang berkaitan dengan permutasi array.
- Berikan contoh lain dari randomized algorithm yang Anda ketahui.
- Pada algoritma masalah *hiring*, jika diasumsikan bahwa calon pegawai datang secara random, tentukan probabilitas bahwa Anda mempekerjakan tepat dua pegawai.