ADS2 — Weekplan Randomized I (Exam Notes & Solutions)

Title	Date	Author	Sources used	Week plan filename
ADS2 — Randomized I Notes & Solutions	2025-10-23 (Europe/ Copenhagen)	ADS2 Copilot	Weekplan images: weekplan-1.png (p.1), weekplan-2.png (p.2); Slides: slides-1.png…slides-7.png; Textbook: Kleinberg & Tardos, *Algorithm Design*, Ch.13 Randomized Algorithms (notably §§13.1, 13.3, 13.5), pp. 708–750	weekplan-1.png; weekplan-2.png

Coverage Table

Weekplan ID	Canonical ID	Title/Label (verbatim)	Assignment Source	Text Source	Status
1	—	Randomized print [w]	weekplan-1.png §1	KT §13.3 (expectation), slides-1.png	Solved
2	—	[w] Expected values	weekplan-1.png §2	KT §13.3 pp. 719–723; slides-2.png	Solved
3	KT §13.5	Analysis of Selection (phase redefined)	weekplan-1.png §3	KT §13.5 pp. 727–733; slides-3.png	Solved
4	—	Christmas party at DTU (exam 2015): Find student with most cookies	weekplan-1.png §4 + Alg.2	KT §13.3 (records/ indicators); slides-4.png	Solved
5	—	Boxes of beer	weekplan-1.png §5 → weekplan-2.png §5.1–5.4	KT §13.3 (geometric/ linearity)	Solved
6*	—	Nuts and bolts (G. J. E. Rawlins)	weekplan-2.png bottom (label shows “5”; treated as 6)	KT §13.5 (quicksort idea); slides-5.png	Solved

MISMATCH note (plan numbering): The week plan lists two items with number “5”. We treat **“Nuts and bolts”** as **Exercise 6** for contiguity. All items 1–6 are covered.

General Methodology and Theory

- **Toolkit:** indicators, linearity $E\left[\sum X_i\right] = \sum E[X_i]$; geometric waiting time ($\mathrm{Geom}(p)$): $E[T] = 1/p$ (failures-before-success variant has $E[F] = (1-p)/p$); union bound; random partition arguments; harmonic numbers $H_n = \sum_{i=1}^{n} 1/i = \Theta(\log n)$.
- **Randomized selection (KT §13.5):** pick a random splitter; expected linear time via phases. With phase threshold ratio $\rho < 1$ and probability $\alpha > 0$ to pick a "central" splitter each iteration, expected iterations/phase $= 1/\alpha$ and total $O\left(\sum_j n\rho^j\right) = O(n)$.
- **Records in a random permutation:** $\Pr[\text{record at position } i] = 1/i$, so $E[\#\text{records}] = H_n$.

Notes

- **Slides-first alignment:** All solutions follow the slide logic for indicators, geometric RVs, selection phases, and randomized quicksort; KT Ch.13 is cited as an alternative view where it differs in constants (e.g., $3/4$ vs $2/3$ phase ratios).
- **Conventions:** Uniform $\mathrm{rand}(1, 10)$ draws are independent; cookies counts are all distinct (as stated); boxes-of-beer has exactly k beer boxes.

Solutions

Exercise 1 — Randomized print [w]

Assignment Source: weekplan-1.png §1 — *Algorithm 1: RandomizedPrint(i)*.
Text Source: KT §13.3 (geometric), slides-1.png.

- Let $p = \Pr[\text{stop on a test}] = \Pr[\text{two equal in } \{1..10\}] = 1/10$. The stars printed equal the number of failures (inequalities) before first success.
- **1.1** $\Pr[\text{exactly 3 stars}] = (1-p)^3\, p = (9/10)^3 \cdot (1/10) = 729/10000.$
- **1.2** $E[\text{stars}] = (1-p)/p = (9/10)/(1/10) = 9.$

Verification: independent trials; geometric model applies; boundary case $p = 1/10$ checked digit-by-digit.
Transfer Pattern: *Geometric waiting time* → cue: "repeat until equality"; mapping: trial=condition check, success=equality.
✅**Answer:** $\Pr[3] = 0.0729;\ E = 9.$

Exercise 2 — [w] Expected values

Assignment Source: weekplan-1.png §2.
Text Source: KT §13.3 pp. 719–723; slides-2.png.

- **2.1** $E[X] = 2 \cdot \frac{1}{3} + 5 \cdot \frac{1}{2} + 8 \cdot \frac{1}{6} = \frac{4}{6} + \frac{15}{6} + \frac{8}{6} = \frac{27}{6} = \frac{9}{2} = 4.5.$
- **2.2** Indicator $I \in \{0, 1\} : E[I] = 0 \cdot \Pr[I = 0] + 1 \cdot \Pr[I = 1] = \Pr[I = 1].$

Transfer Pattern: *Linearity of expectation* → cue: mixture of discrete values or indicators.
✅**Answer:** $E[X] = 4.5;\ E[I] = \Pr[I = 1].$

Exercise 3 — KT §13.5 — Analysis of Selection (phase redefined)

Assignment Source: weekplan-1.png §3.
Text Source: KT §13.5 pp. 727–733; slides-3.png.

- **Phase definition (plan):** phase j has size in $\left(n(2/3)^j,\ n(2/3)^{j+1}\right]$. Call a splitter **central** if at least a third of elements lie on each side.
- **Probability of central:** exactly $n/3$ ranks (between $\lceil n/3 \rceil$ and $\lfloor 2n/3 \rfloor$) $\to \alpha = 1/3$.
- **Per-phase iterations:** geometric with $p = \alpha \Rightarrow E[\text{iters/phase}] = 1/p = 3$.
- **Per-iteration work in phase j :** $O\left(n(2/3)^j\right)$.
- **Total expected time:** $\sum_{j\geq 0} 3 \cdot c\, n(2/3)^j = 3cn \cdot \frac{1}{1-2/3} = 9cn = O(n)$.

Alternative Approach (KT): Using $3/4$-central elements gives factor 2 iters/phase; same linear bound.
Pitfalls: redefining "central" inconsistently; forgetting independence across iterations.
Transfer Pattern: *Randomized selection* → cues: "random splitter", "expected linear"; mapping: choose splitter → partition → recurse on one side.
✅**Answer: Yes.** With the $2/3$ phase definition, expected time remains $O(n)$.

Exercise 4 — Christmas party at DTU (exam 2015)

Assignment Source: weekplan-1.png §4 + Algorithm 2.
Text Source: KT §13.3 (records), slides-4.png.

Algorithm (from plan): scan students in a **random order**; update at line (∗) when you see a new maximum.

- **4.1** $\Pr[(*) \text{ executes at last iteration}] = \Pr[\text{last is global max}] = 1/n$.
- **4.2** Let $X_i = \mathbf{1}[(*) \text{ executes at iteration } i]$. Then $\Pr[X_i = 1] = 1/i$ (position i is a record w.p. $1/i$).
- **4.3** $E\left[\#(*)\right] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} 1/i = H_n$.

Verification: distinct cookie counts ⇒ total order; random permutation symmetry.
Transfer Pattern: *Records in permutations* → cues: "random order, count maxima updates".
✅**Answer:** (4.1) $1/n$; (4.2) $1/i$; (4.3) H_n.

Exercise 5 — Boxes of beer (n boxes, k with beer)

Assignment Source: weekplan-1.png §5 and weekplan-2.png §5.1–5.4.
Text Source: KT §13.3; slides-5.png.

Deterministic baseline (open B1, B2, … until first beer):

- **5.1 Best-case:** 1 (B1 has beer).
- **5.2 Worst-case:** $n - k + 1$ (all $n - k$ empties first, then first beer). Assumes $k \geq 1$.

Randomized v1 (with replacement): *Each trial picks a uniform i∈{1..n}, independent, until a beer is found.*

- **5.3 Expected time:** geometric with success $p = k/n \Rightarrow E[T] = n/k$. Worst case (over randomness): **unbounded** (no finite upper bound on trials).

Randomized v2 (without replacement): *Each round pick a previously unopened box uniformly; stop at first beer.* Let **E** be the set of empties, $|E| = n - k$. For each empty $e \in E$, define indicator $X_e = \mathbf{1}[e \text{ opened before any beer}]$; total opened $X = 1 + \sum_{e \in E} X_e$.

- **5.4.1 Worst-case:** still $n - k + 1$ (open all empties first).
- **(i) Express X :** $X = 1 + \sum_{e \in E} X_e$.
- **(ii) $E[X_e]$:** Among the $k + 1$ boxes (the empty e plus all k beer boxes), the earliest in the random order is uniform $\Rightarrow E[X_e] = \Pr[X_e = 1] = 1/(k + 1)$.
- **(iii) $E[X]$:** $E[X] = 1 + \sum_{e \in E} E[X_e] = 1 + (n - k) \cdot \frac{1}{k+1} = \frac{n+1}{k+1}$.
- **(iv) Expected running time:** $\boxed{(n + 1)/(k + 1)}$.

Pitfalls: mixing “with” vs “without” replacement; forgetting the +1 for the successful beer opening.
Transfer Pattern: *Search until first marked item* → cues: “k successes in n positions”, “random order”.
✅**Answer:** Best 1 , worst $n - k + 1$; randomized-with-replacement: $E = n/k$, worst unbounded; randomized-without-replacement: $E = (n + 1)/(k + 1)$, worst $n - k + 1$.

Exercise 6 — Nuts and bolts (G. J. E. Rawlins)

Assignment Source: weekplan-2.png (bottom).
Text Source: Slides & KT §13.5 (quicksort idea).

Problem. Match each of N nuts to its unique bolt when only *cross* comparisons are allowed (you may compare nut vs bolt; never nut vs nut or bolt vs bolt).

Method (quicksort-style partitioning):

```
Algorithm: match_nuts_and_bolts
Input: sets NUTS={n1..nN}, BOLTS={b1..bN}; comparator cmp(n, b) ∈ {<,=,>}
Output: matched pairs (n_i ↔ b_j) for all i

procedure solve(NUTS, BOLTS):
  if |NUTS| ≤ 1: return NUTS paired with BOLTS
  pick pivot nut n★ uniformly at random
  // partition bolts using n★
  B<, B=, B> ← partition BOLTS by cmp(n★, b)
  let b★ be the unique bolt in B=
  // partition nuts using b★
  N<, N=, N> ← partition NUTS by cmp(n, b★)
  // recurse on corresponding sides
  solve(N<, B<); output (n★, b★); solve(N>, B>)
```

```
return
// Time (expected): O(N log N); Space: O(log N) recursion
```

Why it works: each partition uses only legal cross-comparisons; pivot nut matches exactly one bolt $b_\star$; subproblems are proper; random pivot gives the usual quicksort recurrence.

Verification: at every step, invariants hold—no nut/bolt crosses to the wrong side; sizes of corresponding partitions match.

Variant Drill: choose bolt pivot first; symmetric algorithm, same bound.
Alternative Approach: deterministic linear-time selection for pivots yields $O(N \log N)$ worst-case (more complex).
Transfer Pattern: *Quicksort partition with cross-type comparator* → cues: "two dual sets, only cross comparisons".
✅**Answer:** Expected running time $O(N \log N)$; all pairs matched.

Puzzle

You draw i.i.d. uniform integers in $\{1, \ldots, 100\}$ until you see a repeat. What is the expected number of draws? *(Hint: birthday process; coupon-collector with stopping at first collision.)*
Answer sketch: $E \approx \sqrt{\frac{\pi}{2} \cdot 100} \approx 12.53$ (Poisson birthday approximation).

Summary

- **Core identities:** indicators + linearity; geometric waiting times; harmonic records H_n .
- **Algorithms:** randomized selection stays **linear** even with $2/3$ phase thresholds; scan-max records problem yields $\Pr[\text{record at } i] = 1/i$; nuts-and-bolts via quicksort partitions → **expected** $O(n \log n)$.
- **Beer boxes:** deterministic worst $n - k + 1$; random with replacement $E = n/k$ (unbounded worst); without replacement $E = (n+1)/(k+1)$.
- **Notation:** $H_n,\ E[\cdot],\ \mathbf{1}[\cdot],\ \rho,\ \alpha.$

Slides are the primary authority; KT Ch.13 cited for background and constants.

Randomized Algorithms

Randomized algorithms

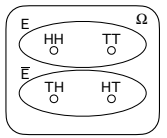
- Today
  - Basic randomized algorithms
  - Expectation of random variables
    - Guessing cards
  - Selection
  - Quicksort

Random Variables and Expectation

Probability

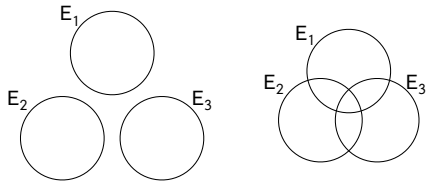
- Probability spaces.
  - Set of possible outcomes Ω.
  - Each item $i \in \Omega$ has probability $\Pr[i] \geq 0$ and $\sum_{i \in \Omega} \Pr[i] = 1$.
  - Event E is a subset of Ω and probability of E is $\Pr(E) = \sum_{i \in E} \Pr[i]$.
  - The complementary event $\overline{E}$ is $\Omega - E$ and $\Pr(\overline{E}) = 1 - \Pr(E)$.

- Example. Flip two fair coins.
  - $\Omega = \{\mathsf{HH}, \mathsf{HT}, \mathsf{TH}, \mathsf{TT}\}$.
  - $\Pr[i] = 1/4$ for each outcome i.
  - Event E = "the coins are the same"
  - $\Pr(\overline{E}) = 1/2$.



Probability

- Union bound.
  - What is the probability that any of event $E_1, \ldots, E_k$ will happen, i.e., what is $\Pr(E_1 \cup E_2 \cup \cdots \cup E_k)$?



  - If events are disjoint, $\Pr(E_1 \cup \cdots \cup E_k) = \Pr(E_1) + \cdots + \Pr(E_k)$.
  - If events overlap, $\Pr(E_1 \cup \cdots \cup E_k) < \Pr(E_1) + \cdots + \Pr(E_k)$.
  - In both cases, the union bound holds:

$$\Pr(E_1 \cup \cdots \cup E_k) \leq \Pr(E_1) + \cdots + \Pr(E_k)$$

Probability

- Independence.
  - Events E and F are independent if information about E does not affect outcome of F and vice versa.
  - Same as $\Pr(E \cap F) = \Pr(E) \cdot \Pr(F)$

Random variables

- A random variable is an entity that can assume different values.
- The values are selected "randomly"; i.e., the process is governed by a probability distribution.
- Examples: Let X be the random variable "number shown by dice".
  - X can take the values 1, 2, 3, 4, 5, 6.
  - If it is a fair dice then the probability that X = 1 is 1/6:
    - Pr[X=1] = 1/6.
    - Pr[X=2] = 1/6.
    - …

Expected values

- Let X be a random variable with values in $\{x_1,\ldots x_n\}$, where x_i are numbers.
- The expected value (expectation) of X is defined as

$$E[X] = \sum_{j=1}^{n} x_j \cdot \Pr[X = x_j]$$

- The expectation is the theoretical average.
- Example:
  - X = random variable "number shown by dice"

$$E[X] = \sum_{j=1}^{6} j \cdot \Pr[X = j] = (1+2+3+4+5+6) \cdot \frac{1}{6} = 3.5$$

Waiting for a first succes

- Coin flips. Coin is heads with probability p and tails with probability $1 - p$. How many independent flips X until first heads?
  - Probability of $X = j$? (first succes is in round j)

$$\Pr[X = j] = (1-p)^{j-1} \cdot p$$

  - Expected value of X:

$$E[X] = \sum_{j=1}^{\infty} j \cdot \Pr[X = j] = \sum_{j=1}^{\infty} j \cdot (1-p)^{j-1} \cdot p = \frac{p}{1-p} \sum_{j=1}^{\infty} j \cdot (1-p)^{j}$$

$$= \frac{p}{1-p} \cdot \frac{1-p}{p^2} = \frac{1}{p}$$

$$\sum_{k=0}^{\infty} k \cdot x^k = \frac{x}{(1-x)^2} \quad \text{for } |x| < 1.$$

Properties of expectation

- If we repeatedly perform independent trials of an experiment, each of which succeeds with probability $p > 0$, then the expected number of trials we need to perform until the first succes is $1/p$.
- If X is a 0/1 random variable, then $E[X] = \Pr[X = 1]$.
- Linearity of expectation: For two random variables X and Y we have

$$E[X + Y] = E[X] + E[Y]$$

Guessing cards

- Game. Shuffle a deck of n cards; turn them over one at a time; try to guess each card.
- Memoryless guessing. Can't remember what's been turned over already. Guess a card from full deck uniformly at random.
- Claim. *The expected number of correct guesses is 1.*
  - $X_i = 1$ if i^{th} guess correct and zero otherwise.
  - X = the correct number of guesses $= X_1 + \ldots + X_n$.
  - $E[X_i] = \Pr[X_i = 1] = 1/n$.
  - $E[X] = E[X_1 + \cdots + X_n] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/n = 1$.

Guessing cards

- Game. Shuffle a deck of n cards; turn them over one at a time; try to guess each card.
- Guessing with memory. Guess a card uniformly at random from cards not yet seen.
- Claim. *The expected number of correct guesses is* $\Theta(\log n)$.
  - $X_i = 1$ if i^{th} guess correct and zero otherwise.
  - X = the correct number of guesses $= X_1 + \ldots + X_n$.
  - $E[X_i] = \Pr[X_i = 1] = 1/(n - i + 1)$.
  - $E[X] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/2 + 1/1 = H_n$.

$$\ln n < H(n) < \ln n + 1$$

Coupon collector

- Coupon collector. Each box of cereal contains a coupon. There are n different types of coupons. Assuming all boxes are equally likely to contain each coupon, how many boxes before you have at least 1 coupon of each type?
- Claim. *The expected number of steps is* $\Theta(n \log n)$.
  - Phase j = time between j and $j+1$ distinct coupons.
  - X_j = number of steps you spend in phase j.
  - X = number of steps in total = $X_0 + X_1 + \cdots + X_{n-1}$.
  - $E[X_j] = n/(n-j)$.
  - The expected number of steps:

$$E[X] = E[\sum_{j=0}^{n-1} X_j] = \sum_{j=0}^{n-1} E[X_j] = \sum_{j=0}^{n-1} n/(n-j) = n \cdot \sum_{i=1}^{n} 1/i = n \cdot H_n.$$

Median/Select

Select

- Given n numbers $S = \{a_1, \ldots, a_n\}$.
- Median: number that is in the middle position if in sorted order.
- Select(S,k): Return the kth smallest number in S.
  - Min(S) = Select(S,1), Max(S)= Select(S,n), Median = Select(S,$n/2$).
- Assume the numbers are distinct.

```
Select(S, k)

   Choose a pivot s ∈ S uniformly at random.

   For each element e in S:
    if e < s put e in S'
    if e > s put e in S''

   if |S'| = k-1 then return s

   if |S'| ≥ k then call Select(S', k)

   if |S'| < k then call Select(S'', k - |S'| - 1)
```

Select: Running time

```
Select(S, k)
   Choose a pivot s ∈ S uniformly at random.

   For each element e in S:
    if e < s put e in S'
    if e > s put e in S''

   if |S'| = k-1 then return s

   if |S'| ≥ k then call Select(S', k)

   if |S'| < k then call Select(S'', k - |S'| - 1)
```

- Worst case running time: $T(n) = cn + c(n-1) + c(n-2) + \ldots + c = \Theta(n^2)$
- If there is at least an ε fraction of elements both larger and smaller than s:

$$\begin{aligned} T(n) &= cn + (1-\varepsilon)cn + (1-\varepsilon)^2 cn + \ldots \\ &= (1 + (1-\varepsilon) + (1-\varepsilon)^2 + \ldots)cn \\ &\leq cn/\varepsilon \end{aligned}$$

- Intuition: A fairly large fraction of elements are “well-centered” => random pivot likely to be good.

Select: Analysis

- Central element: ≥ 1/4 of the elements in current S are smaller and ≥ 1/4 are larger.

- If pivot central: size of set shrinks by at least a factor 3/4.
- At least half the elements are central ⇒ Pr[s is central] = 1/2.
- Phase j: Size of set at most $(3/4)^j n$ and at least $(3/4)^{j+1} n$.
  - Pivot central ⇒ current phase ends.
  - Expected number of iterations before a central pivot is found = 2.
- X = number of steps taken by algorithm. X_j = number of steps in phase j.
- Then $X = X_0 + X_1 + X_2 + \cdots$
- $E[X_j] = 2cn(3/4)^j$
- Expected running time:

$$E[X] = E[\sum_j X_j] = \sum_j E[X_j] = \sum_j 2cn \left(\frac{3}{4}\right)^j = 2cn \sum_j \left(\frac{3}{4}\right)^j \leq 8cn$$

Quicksort

Quicksort

- Given n numbers $S = \{a_1, \ldots, a_n\}$.
- Assume the numbers are distinct.

```
Quicksort(S)

  if |S| ≤ 1 return S

  else

   Choose a pivot s ∈ S uniformly at random.

   For each element e in S
     if e < s put e in S'
     if e > s put e in S''

  L = Quicksort(S')
  R = Quicksort(S'')

  return the sorted list L∘s∘R
```

Quicksort

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	19	3	1	7	8	17	9	2	13	23	10	25	24	16	21	11	14	12	15

Quicksort

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	19	3	1	7	8	17	9	2	13	23	10	25	24	16	21	11	14	12	15

Quicksort

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	19	3	1	7	8	17	9	2	13	23	6	25	24	16	21	11	14	12	15
5	3	1	7	8	2														
19	17	13	23	10	25	24	16	21	11	14	12	15							

Recurse!

Quicksort

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	19	3	1	7	8	17	9	2	13	23	6	25	24	16	21	11	14	12	15
1	2	3	5	7	8														
10	11	12	13	14	15	16	17	19	21	23	24	25							

Combine!

Quicksort

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	19	3	1	7	8	17	9	2	13	23	6	25	24	16	21	11	14	12	15
1	2	3	5	7	8														
10	11	12	13	14	15	16	17	19	21	23	24	25							
1	2	3	5	7	8	9	10	11	12	13	14	15	16	17	19	21	23	24	25

Quicksort: Analysis

- Worst case: $\Omega(n^2)$ comparisons.
- Best case: $O(n \log n)$
- Enumerate elements such that $a_1 \leq a_2 \leq \cdots \leq a_n$.
- Indicator random variable for all pairs $i < j$:

$$X_{ij} = \begin{cases} 1 & \text{if } a_i \text{ and } a_j \text{ are compared by the algorithm} \\ 0 & \text{otherwise} \end{cases}$$

- X total number of comparisons:

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

- Expected number of comparisons:

$$E[X] = E[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$$

Quicksort: Analysis

- Compute expected number of comparisons.
- Since X_{ij} is an indicator variable: $E[X_{ij}] = \Pr[X_{ij} = 1]$.
- a_i and a_j compared $\Leftrightarrow$ a_i or a_j is the first pivot element chosen from $Z_{ij} = \{a_i, \ldots, a_j\}$
- Pivot chosen independently uniformly at random $\Rightarrow$

  all elements from Z_{ij} equally likely to be chosen as first pivot from this set.
- We have $\Pr[X_{ij} = 1] = 2/(j - i + 1)$.
- Thus

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr[X_{ij} = 1] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}$$

$$= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \leq \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} = 2 \sum_{i=1}^{n-1} H_n = 2n \cdot H_n \leq O(n \log n)$$

1 Randomized print $[w]$ Consider the following program:

Algorithm 1: RandomizedPrint(i)

```
while rand(1, 10) ≠ rand(1, 10) do
    print(∗);
end
```

1.1 What is the probability that exactly 3 stars are printed?

1.2 What is the expected number of stars that are printed?

2 $[w]$ **Expected values**

2.1 Let X be a random variable which assumes the values 2, 5 and 8 with probabilities 1/3, 1/2 and 1/6 respectively, i.e., $\Pr[X = 2] = 1/3$ etc. What is the expected value of X?

2.2 An *indicator random variable* is a random variable that only assumes the values 0 and 1. Prove that for an indicator random variable X we have $E[X] = \Pr[X = 1]$.

3 Analysis of Selection Instead of defining a phase as in the book we now define a phase like follows: The algorithm is in *phase j* when the size of the set under consideration is at most $n(\frac{2}{3})^j$ but greater than $n(\frac{2}{3})^{j+1}$. Do the analysis of Selection with this definition of phase. Do you still get a linear bound on the expected running time?

4 Christmas party at DTU (exam 2015) During the Christmas party at DTU party the Dean suddenly wants to know who won most Christmas cookies in the "Bing or Ding" game. He suggests the following algorithm:

Algorithm 2: Find student with most cookies

```
max ← −∞
s ←null
Randomly order the students. Let s_1, ..., s_n be the students in this random order.
Let c_i denote the number of cookies won by student s_i.
for i = 1, ..., n do
    if if c_i > max then
        max ← c_i and s ← s_i    (*)
    end
end
return s
```

In the following assume that all students won a different amount of cookies. That is, $c_i \neq c_j$ for all $i \neq j$.

4.1 What is the probability that the line (*) is executed at the last iteration?

4.2 Let X_i be a random variable that is 1 if line (*) is executed in iteration i and 0 otherwise. What is the probability that $X_i = 1$?

4.3 What is the expected number of times line (*) is executed?

5 Boxes of beer You are given n boxes $B_1, \ldots, B_n$. Exactly k boxes contain a bottle of beer ($k \leq n$) the rest is empty. From the outside one cannot see whether a box is empty or not. The aim is to find a box with a beer it. The following deterministic algorithm is suggested: Open the boxes $B_1, B_2, \ldots$ in this order. The algorithm stops when a beer is found. We count opening a box as one computational step.

5.1 What is the best-case running time of the deterministic algorithm.

5.2 What is the worst-case running time of the deterministic algorithm.

5.3 Consider the following randomized algorithm: Randomly pick a number $i \in \{1, 2, \ldots, n\}$ and open box B_i.

1. What is the expected running time of this algorithm?
2. What is the worst-case running time of this algorithm?

5.4 Now consider a modification of the randomized algorithm above: Randomly pick a box you have not opened before and open it.

What is the worst case running time of this algorithm?

We now want to analyse the expected running time of this algorithm. Make a indicator variable X_i for each empty box i, where $X_i = 1$ if box i was opened by the algorithm and 0 otherwise. Let X be the number of boxes opened by the algorithm.

1. Express X using the X_is, i.e., $X =$ .....
2. What is the expected value of X_i?
3. What is the expected value of X (use the bounds on the expected value of the X_is)?
4. What is the expected running time of the algorithm?

5 Nuts and bolts. (G. J. E. Rawlins) You have a mixed pile of N nuts and N bolts and need to quickly find the corresponding pairs of nuts and bolts. Each nut matches exactly one bolt, and each bolt matches exactly one nut. By fitting a nut and bolt together, you can see which is bigger. But it is not possible to directly compare two nuts or two bolts. Given an efficient method for solving the problem.[1]
Hint: customize quicksort to the problem. Side note: only a very complicated deterministic $O(N \log N)$ algorithm is known for this problem.

[1] This exercise is from http://algs4.cs.princeton.edu/23quicksort/