Advanced Algorithms and Data structures
Assignment three

Magnus Goltermann (xzb187), Thomas Busk-Jepsen (tnr653)
Mia Rahlff Pedersen(bvx284)

December 9, 2024

1 2.1

Yes the truly independent hash function $h : U \to [m]$ is universal. As the mapping is independent and universal the chance of having a collision is exactly $\frac{1}{m}$ meaning that it is universal.

2 2.2

If we either have an infinitely large m then the probability will in the limit go towards 0. Or any m given we have an empty U.

3 2.3

The identity function $f(x) = x$ is an universal hash function $[u] \to [m]$ as the mapping from two distinct values in U always maps to different values as the hash values are larger than the universe. Therefore the chance of collision is less than $\frac{1}{m}$ as it is 0

4 2.4

4.1 a

We, first of all, know that we always have a length of at least 1, since x is given in to be in L. Then we know that we have $n-1$ more elements, which all have a probability of $\frac{1}{m}$ hit the same key. And thus if we combine it we have

$$\mathbb{E}[L(h(x))] = 1 + \sum^{n-1} \frac{1}{m} = 1 + \frac{n-1}{m} \tag{1}$$

4.2 b

As we have that h is 2-approximately universal, we have that

$$\Pr_h[h(x) = h(y)] \leq \frac{2}{m}$$

Substituting this in, in following equation, As we may now assume that x is not in S

$$E_h[\|L[h(x)]\|] = E_h\left[\sum_{y \in S} I(y)\right] = \sum_{y \in S} E_h[I(y)] = \sum_{y \in S} \Pr_h[h(y) = h(x)] \leq n/m \leq 1. \tag{2}$$

We get

$$\mathbb{E}[|L[h(x)]|] \leq \sum_{y \in S} \frac{2}{m} = n \cdot \frac{2}{m} = \frac{2n}{m} \tag{3}$$

5 2.5

As we know that both hash functions are universal then we know that a probability of a collision is at most $\frac{1}{m}$ Where m is the size of distinct hash values. Therefore we know that the chances of $h(x) = h(y) = \frac{1}{n}$ for the first and $h(x) = h(y) = \frac{1}{n^3}$. We can then take the joint probability and then we get: $\frac{1}{n} \cdot \frac{1}{n^3} = \frac{1}{n^4}$ we then have to do it for all of S and as $|S| = n$ We get $\frac{n}{n^4} = \frac{1}{n^3}$

6 2.6

6.1 a

We know, that for a function to be universal, we must have that

$$\Pr_h[h(x) = h(y)] \leq \frac{1}{m}$$

When we set $a = 0$, we have that

$$h_{0,b}(x) = (b \bmod p) \bmod m$$

We thereby have, that the hash function does not depend on x, solely on b. This gives us the probability of $h_{0,b}(x) = h_{0,b}(y)$ is 1. This exceeds, the limit of $\frac{1}{m}$, making it not universal.

6.2 b

Given the normal case when not taking $a = 0$ into account, we know that the probability of collision is $\frac{1}{m}$, and from the previous task we know that when

$a = 0$ there is a 1 probability of collision, but $Pr[a = 0] = \frac{1}{p}$, and thus the likelihood of collision must be

$$Pr[h_{a,b}(x) = h_{a,b}(y)] = Pr[a = 0] \cdot 1 + Pr[a \neq 0] \cdot \frac{1}{m}$$

And we know $Pr[a = 0] = \frac{1}{p}$, and thus $Pr[a \neq 0] = 1 - Pr[a = 0] = 1 - \frac{1}{p}$

$$Pr[a = 0] \cdot 1 + Pr[a \neq 0] \cdot \frac{1}{m} = \frac{1}{p} + (1 - Pr[a = 0])\frac{1}{m} = \frac{m + p - 1}{mp}$$

Since we know that $p \geq m$ we have

$$\frac{m + p - 1}{mp} < \frac{2}{m}$$

7 3.1

For 3-indepencence it must then be the independence of $h(x)$, $h(y)$, and $h(z)$ for the distinct keys $x, y, z \in U$ and for the distinct hash values $q, r, s \in [m]$ we have the probability that those 3 keys map to 3 different values is

$$Pr[h(x) = q \wedge h(y) = r \wedge h(z) = s] = Pr[h(x) = q] \cdot Pr[h(y) = r] \cdot Pr[h(z) = s] \tag{4}$$

And given they're all uniformly distributed we have each probability to $\frac{1}{m}$, and thus

$$(4) = \frac{1}{m^3} \tag{5}$$

Following the same principle for k-independence, we simply extend the same argument, and thus end up with

$$Pr[h(x_1) = q_1 \wedge h(x_2) = q_2 \wedge \cdots \wedge h(x_k) = q_k] = \frac{1}{m^k} \tag{6}$$

8 3.2

Since the two hash independently as given in the definition, we have that

$$Pr[h(x) = q \wedge h(y) = r] = Pr[h(x) = q] \cdot Pr[h(y) = r] \tag{7}$$

And since both are independently bounded by $\frac{c}{m}$, we have

$$Pr[h(x) = q] \cdot Pr[h(y) = r] \leq \frac{c^2}{m^2} \tag{8}$$

9 3.3

If h is c-approximately strongly universal, we have that

$$\Pr[h(x) = q \wedge h(y) = q] \leq \frac{c}{m^2} \tag{9}$$

Substituting in

$$\Pr[h(x) = h(y)] = \sum_{q \in [m]} \Pr[h(x) = q \wedge h(y) = q]. \tag{10}$$

We get

$$\Pr[h(x) = h(y)] \leq \sum_{q \in [m]} \frac{c}{m^2} \tag{11}$$

Simplifying to

$$\Pr[h(x) = h(y)] \leq \frac{c}{m} \tag{12}$$

10 3.4

For the multiply-shift to be c-approximately strong universal for any constant c, it would have to has all pairs of keys to be independent. But as a counter-example, we see that if we have

$$y - x = 2^{w-l}$$

Where x and y are keys, then after the multiplication of a we have

$$a(y - x) = a2^{w-l}$$

This means we can get a deterministic difference between the two values, and thus not all pairs of keys hash independently, making it not c-approximately strong universal.

11 3.5

First, we see that the set they give are all the elements in B and C which are not in both, thus its everything else than their intersection

$$B \cup C \backslash B \cap C$$

the size of that can be computed as the size of the B and C subtracted with twice the intersection (since it is represented twice):

$$|B \cup C \backslash B \cap C| = |B| + |C| - 2|B \cap C|$$

And we know that the sizes can be estimated as

$$|B| = S_{h,t}(B) \cdot \frac{m}{t}$$

$$|C| = S_{h,t}(C) \cdot \frac{m}{t}$$

$$|B \cap C| = S_{h,t}(B) \cap S_{h,t}(B) \cdot \frac{m}{t}$$

And thus the estimated size of the symmetric difference of B and C is

$$S_{h,t}(B) \cdot \frac{m}{t} + S_{h,t}(C) \cdot \frac{m}{t} - 2S_{h,t}(B) \cap S_{h,t}(B) \cdot \frac{m}{t}$$

12 3.6

Using lemma 3.2 we get the equation

$$Pr[|X - 1.000.000| \geq q \cdot \sqrt{1.000.000}] \leq \frac{1}{q^2} \tag{13}$$

which gives and upper bound of the probability, thus we just need to solve for q. And since we know that $|X - 1.000.000| \geq 10.000$, we can just solve $q \cdot \sqrt{1.000.000} = 10.000 \iff q = 10$, and thus the bound ens up with

$$Pr[|X - 1.000.000| \geq 10.000] \leq \frac{1}{10^2} = \frac{1}{100} \tag{14}$$

13 34.1-1

If LONGEST-PATH is in p, and we know that the longest path can never exceed the number of vertices or edges, a naive approach could be to just do a linear search from $0 \ldots |V|$ trying all values until LONGEST-PATH, and thus we find the longest path in $O(|V| \cdot p(n))$ where $p(n)$ is the polynomial time complexity of LONGEST-PATH. We could also do a more efficient solution in $O(log(|V|) \cdot p(n))$ if we use a binary search instead.

14 34.1-5

If we have an algorithm that does a constant number of polynomial time operations and another polynonimal time, it can be expressed as just

$$C \cdot p(n) + p(n)$$

Where C is a constant and $p(n)$ is a polynomial, thus we need that

$$0 \leq C_1 \cdot p(n) + p(n) \leq C_2 \cdot p(n)$$

If we simply pick C_2 in terms of C_1 we can rewrite it to be guaranteed to hold

$$0 \leq (C_1 + 1) \cdot p(n) \leq (C_1 + 2) \cdot p(n) \tag{15}$$

and thus $O(p(n))$.
If algorithm A calls algorithm B n times, and both algorithms run in some $p(n)$

time, but every time B is called, the input size is e.g. doubled, then we would end with a combined algorithm with a time complexity of:

$$T(n) = \sum_{i=0}^{m-1} (2^i n)^k$$

Where k is the degree of the polynomial of $p(n)$. Rewriting the sum:

$$\sum_{i=0}^{m-1} (2^i n)^k = n^k \sum_{i=0}^{m-1} (2^{ik}) \tag{16}$$

Which is a geometric sum, which gives

$$n^k \sum_{i=0}^{m-1} (2^{ik}) = n^k \frac{2^{nk} - 1}{2^k - 1} = O(2^n) \tag{17}$$

Thus it has an exponential time complexity.

15 34.2-5

Since we know that $L \in NP$, then we also know that the solution can be verified with some polynomial-time algorithm. A certificate length is bound by some polynomial $p(n)$, and if such a certificate is made into a binary string, the number of certificates can be bound by looking at all combinations of binary strings (certificates), which have a length equal to and below the length of the certificate ($2^{p(n)}$). Therefore we end up looking over all possible certificates, which will take $2^{O(n^k)}$.

16 34.2-6

First, we set up a certificate for the solution to be the sequences P of vertices that give a Hamiltonian path in the graph G:

$$P = [u, v_1, v_2, \ldots, v_k, v] \tag{18}$$

Where u is the starting vertex and v is the last vertex in the path. We further know that the length of the sequence P cannot be longer than $|V|$, since it contains all the vertices exactly once, and thus we can iterate through and check them in $O(|V|)$ time. And since we can verify a solution in polynomial time, `HAM-PATH` belongs to NP.

17 34.2-8

The complement of a tautology is a formula in which at least one combination of the variables exists, which makes the formula evaluate to false. A certificate

for the formula in the compliment tautology would just be what the variables should be set to for the formula to evaluate to false, which will take $O(n)$ to verify since we need to evaluate all the expressions in the formula for that set of variable values. And since the compliment then can be verified in polynomial time, thus being in NP, and then the tautology is in co-NP.

18 34.3-2

Given the reduction function $f_1(x)$ which reduces L_1 to L_2, and the reduction function $f_2(x)$ which reduces L_2 to L_3, then must $f_2(f_1(x))$ reduce L_1 to L_3.

19 34.3-3

To prove that we can reduce L to $\bar{L}$ if and only if we also can reduce $\bar{L}$ to L. Thus we need to prove that when L can be reduced to $\bar{L}$, then $\bar{L}$ can also be reduced to L:
There must exist a function f that:

$$x \in L \iff f(x) \in \bar{L}$$

If we assume that the above is true such that $L \leq_p \bar{L}$, then we just need to show that $\bar{L} \leq_p L$ such that

$$x \in \bar{L} \iff f(x) \in L$$

since $x \in \bar{L}$, then $x \notin L$ and by definition $f(x) \in L$. And if $x \notin \bar{L}$, then $x \in L$ and thus $f(x) \notin L$. Doing the other direction is the same argument just the opposite, and thus we have that we can reduce L to $\bar{L}$ if and only if we also can reduce $\bar{L}$ to L.