Solution 1

(a) Let $C$ be an almost minimum cut of size $K$. First, we prove that the probability that $C$ survives (none of its edges are contracted) until the remaining graph has 4 nodes is bounded by $\Omega(1/n^4)$.

Notice that every node of $G$ has degree at least $\mu(G)$, and thus $|E(G)| \geq n\mu(G)/2$, where $\mu(G)$ stands for the minimum size of a cut in $G$. The probability that $C$ survives the first round is at least

$$1 - \frac{K}{|E(G)|} \geq 1 - \frac{K}{n\mu(G)/2} \geq 1 - \frac{4}{n}$$

where we used $K \leq 2\mu(G)$.

Recall, $\mu(G/e) \geq \mu(G)$ for any edge $e$ in $G$, i.e., contracting an edge does not decrease the minimum size of a cut. Therefore with a similar argument as above, running the algorithm BasicMinCut($G$) until we have at most 4 nodes, the probability that $C$ survives is at least

$$\left(1 - \frac{4}{n}\right)\left(1 - \frac{4}{n-1}\right)\left(1 - \frac{4}{n-2}\right)\left(1 - \frac{4}{n-3}\right)\cdots\left(1 - \frac{4}{5}\right) = \frac{n-4}{n-1}\cdot\frac{n-5}{n-2}\cdot\frac{n-6}{n-3}\cdot\frac{n-7}{n-4}\cdot\frac{1}{5} = \Omega(1/n^4).$$

Now, we have at most 4 nodes and a multi-graph on 4 nodes has constantly many cuts. Therefore, the probability that the algorithm picks $C$ is lower-bounded by a constant. This implies that the algorithm returns $K$ with probability $\Omega(1/n^4)$.

(b) Consider the following multigraph, $G = (V, E)$. Let $V = \{s, t, v_1, \ldots, v_n\}$ and $E = E_S \cup E_T$ where $E_S = \{(s, v_i) : i = 1, \ldots, n\}$ and $E_T = \{(v_i, t) : i = 1, \ldots, n\}$ and the edges in $E_T$ occur twice. The minimum $s$-$t$ cut consists of the node $s$ and has value $n$. So there is one edge from $s$ to each node $v_i$ for all $i$, and two edges from $v_i$ to $t$ for each $i$. Then, in order for the algorithm to terminate successfully, only edges from $E_T$ can be contracted. If an edge from $E_S$ is contracted, the cut value returned will be at least $n + 1$, as neither of the two edges from the contracted node can be chosen (as the contracted node is now the $s$-node). By this same argument, every time an edge is contracted, three edges (including the one contracted) are no longer possible choices as they are either contracted or edges connecting the $s$-node to the $t$-node. Thus, at every iteration the probability of choosing a contraction preserving the possibility of success is $\frac{2}{3}$, and there are a total of $n$ iterations. Then the probability of success is $\left(\frac{2}{3}\right)^n$. 


Solution 2

(a) We give a proof by induction of the following (stronger) statement, which then immediately implies the desired result.

Claim 1. Let $S = \{1, 1+1, \ldots, r\}$ be any set of keys consisting of consecutive natural numbers, and let $j < i$ be two specific keys from that set. Consider the corresponding random search tree $\tilde{B}_S$ (as defined on the exercise sheet) and the random variable $A^i_j$. Then,

$$\Pr[A^i_j = 1] = \frac{2j}{(i-j+1)(i+j)}.$$ 

Proof. The induction is over the size $n := r - 1 + 1$ of the set $S = \{1, 1+1, \ldots, r\}$. We consider the following two cases, one being straightforward and the other inductive.

- $i - j + 1 = n$ (in words, $j = 1$ is the smallest key in $S$ and $i = r$ is the largest). Clearly, in this case $j$ is an ancestor of $i$ if and only if $j$ is the root of the whole tree $\tilde{B}_S$. By definition, this happens with the following probability.

$$\Pr[A^i_1 = 1] = \Pr[\text{"$j$ is root in $\tilde{B}_S$"}] = \frac{j}{\sum_{y \in S} y} = \frac{j}{(\text{r}\text{t}\text{h} + 1)} = \frac{2j}{(i-j+1)(i+j)}$$

- $i - j + 1 < n$ (in words, there are keys in $S$ that are smaller than $j$ or bigger than $i$). In this second case, we use the law of total probability and condition on the event $E$ that the root of $\tilde{B}_S$ is in the range $j, j+1, \ldots, i$.

$$\Pr[A^i_j = 1] = \Pr[A^i_j = 1 \mid E] \cdot \Pr[E] + \Pr[A^i_j = 1 \mid E^c] \cdot \Pr[E^c] = \frac{2j}{(i-j+1)(i+j)}$$

Equality (1) follows for the same reasons as in the first case; conditioned on $E$, $j$ is an ancestor of $i$ if and only if $j$ is the root of $\tilde{B}_S$ and thus we get the same probability. Equality (2) is where we use Claim 1 inductively: for example, if the root of $\tilde{B}_S$ is equal to some key $k < j$, then we apply Claim 1 to the right subtree of $\tilde{B}_S$, which is defined over the (strictly smaller) set of keys $S^{-k} = \{k + 1, k + 2, \ldots, r\}$; analogously, for $k > i$ we apply Claim 1 to left subtree of $\tilde{B}_S$, which is defined over the (strictly smaller) set of keys $S^{-k} = \{1, 1+1, \ldots, k-1\}$. \hfill \Box

(b) We can write the depth of the largest key as $D^{(n)}_n = \sum_{j=1}^{n-1} A^i_j$. By using linearity of expectation and part (a), we then have

$$E[D^{(n)}_n] = \sum_{j=1}^{n-1} E[A^i_j] = \sum_{j=1}^{n-1} \Pr[A^i_j = 1] = \sum_{j=1}^{n-1} \frac{2j}{(n-j+1)(j+n)}$$

Having split the sum in three, we can now analyze all three parts in isolation. The first part is easy to express as a harmonic series, as follows.

$$\sum_{j=1}^{n-1} \frac{1}{n-j+1} = H_n - 1$$
For the second part, we make use of the fact $H_n = \ln(n) + \gamma + o(1)$ from page 24 of the lecture notes, where $\gamma$ is Euler’s constant.\footnote{For those unfamiliar with little-oh notation: Here, it suffices to know that $o(1)$ denotes a term that approaches zero as $n$ tends to infinity.}

\[
\sum_{j=1}^{n-1} \frac{1}{j+n} = H_{2n} - H_n - \frac{1}{2n} = (\ln(2n) + \gamma + o(1)) - (\ln(n) + \gamma + o(1)) - \frac{1}{2n} = \ln(2) + o(1) - \frac{1}{2n} = \ln(2) + o(1)
\]

For the third part, we will do another partial fraction decomposition.

\[
\sum_{j=1}^{n-1} \frac{1}{(n-j+1)(j+n)} = \sum_{j=1}^{n-1} \left( \frac{1}{2n+1} \left( \frac{1}{n-j+1} + \frac{1}{j+n} \right) \right) = \frac{1}{2n+1} ((H_n - 1) + (H_{2n} - H_n)) = \Theta(\ln(n)) = o(1)
\]

Putting everything together, we get the following estimate.

\[
\mathbb{E}[D_n^{[n]}] = H_n - 1 - \ln(2) + o(1)
\]

Up to the constant $\ln(2)$, this is the same as for usual search trees.

**Solution 3**

(a) An example is given as follows:

\[
\begin{array}{c}
q \\
\end{array}
\]

Assume that we have $X_2 = 0$. Then $s_1$ must be the long segment, and $X_3 = 0$. In summary,

\[
\Pr [X_3 = 0 \mid X_2 = 0] = 1.
\]

However, $\Pr [X_3 = 0] < 1$; hence the variables $X_2, X_3$ are not independent.

(b) For each $i = 1, \ldots, n$, let $S_i = \{s_1, \ldots, s_i\}$ denote the arrangement of segments at time $i$. Furthermore, let $C_i \subseteq S_i$ denote the (random) set of segments that are incident to the trapezoid of $S_i$ that contains $q$. Note that $X_i$ is just the indicator variable for the event “$s_i \in C_i$”.

As discussed in the lecture, we have $|C_i| \leq 4$. Now we choose sets $\hat{C}_i$ with the following properties:

- $C_i \subseteq \hat{C}_i \subseteq S_i$, and
- $|\hat{C}_i| = \min \{4, |S_i|\}$. 

\[
\begin{array}{c}
\end{array}
\]
We obtain these sets by simply filling up each $C_i$ with arbitrary elements from $S_i$.) Now we define

$$Y_i = \begin{cases} 1, & \text{if } s_i \in \hat{C}_i, \\ 0, & \text{otherwise.} \end{cases}$$

We have $Y_i \geq X_i$ by construction, and with the abbreviation $p_i = \min \left\{ 1, \frac{4}{i} \right\}$ we find

$$E[Y_i] = \Pr [Y_i = 1] = \Pr [s_i \in \hat{C}_i] = \frac{|\hat{C}_i|}{|S_i|} = p_i.$$

For mutual independence we need to argue that for all choices of $y_1, \ldots, y_n \in \{0, 1\}$ we have

$$\Pr [Y_1 = y_1, \ldots, Y_n = y_n] = \left( \prod_{i : y_i = 1} p_i \right) \left( \prod_{i : y_i = 0} (1 - p_i) \right).$$

(1)

Since the argument will be the same for every choice of the $y_i$'s, and in order to reduce clutter in our notation, we only prove

$$\Pr [Y_1 = 1, \ldots, Y_n = 1] = \prod_{i=1}^{n} p_i.$$  (2)

To this end we employ backwards analysis: Consider the backwards process that starts from the complete arrangement of segments $S$, then removes segments in the order $s_n, s_{n-1}, \ldots, s_1$. At the first step of this process, we remove the element $s_n$ which is chosen uniformly at random from $S$, and our success probability at this step is

$$\Pr [Y_n = 1] = p_n.$$  

At the next step of the backwards process, $s_{n-1}$ is chosen uniformly at random from the set $S \setminus \{s_n\}$. Now by definition, $\hat{C}_{n-1}$ is a 4-element subset of $S_{n-1}$, and we have $S_{n-1} = S \setminus \{s_n\}$. (Crucially, all of this holds independently of the choice of $s_n$.) Hence, our success probability at this step is

$$\frac{|\hat{C}_{n-1}|}{|S \setminus \{s_n\}|} = p_{n-1}.$$  

Continuing this argument, we obtain an overall success probability of $\prod_{i=1}^{n} p_i$, which proves (2).

(c) We have

$$E \left[ e^{\ln(5/4)Y} \right] = E \left[ \left( \frac{5}{4} \right)^Y \right].$$

By independence, this is $\prod_{i=1}^{n} E \left[ \left( \frac{5}{4} \right)^{Y_i} \right]$. We have

$$E \left[ \left( \frac{5}{4} \right)^{Y_i} \right] = 1 \Pr [Y_i = 0] + \frac{5}{4} \Pr [Y_i = 1] = 1 + \frac{1}{4} \Pr [Y_i = 1] \leq 1 + \frac{1}{4} \cdot \frac{i+1}{i} = \frac{i+1}{i}.$$  

Hence

$$\prod_{i=1}^{n} E \left[ \left( \frac{5}{4} \right)^{Y_i} \right] \leq \prod_{i=1}^{n} \frac{i+1}{i} = n + 1.$$  

We have

$$\Pr [Y \geq \lambda \ln(n + 1)] = \Pr \left[ e^{\ln(5/4)Y} \geq e^{\ln(5/4)\lambda \ln(n + 1)} \right].$$
Markov's inequality tells us that that $\Pr[W \geq \alpha] \leq \frac{E[W]}{\alpha}$ for $\alpha > 0$ and $W$ a nonnegative random variable. Setting $W = e^{\ln(5/4)Z}$ and $\alpha = e^{\ln(5/4)\ln(n+1)}$, the above probability is at most

$$e^{-\ln(5/4)\ln(n+1)}(n + 1) = \frac{n + 1}{(n + 1)^{\ln(\frac{5}{4})n}} = \left(\frac{1}{n + 1}\right)^{\ln(5/4) - 1}.$$  

(d) There are $2n$ segment endpoints. Extending these endpoints by passing a vertical line through them partitions the plane in $2n + 1$ regions. Intersecting these regions with all the segments gives us at most $(n + 1)(2n + 1) \leq 2(n + 1)^2$ trapezoids (equivalence classes), such that the points inside each trapezoid behave identically in all possible history graphs. Let $Q$ be a set of query points formed by taking one representative from each equivalence class, and for each $q \in Q$ define the random variable $Y_q$ to be the query time for the query point $q$. Furthermore, let $\lambda$ be such that $\lambda \ln(5/4) - 1 = 3$. Now we can compute that

$$\Pr[\text{query time is less than } \lambda \ln(n + 1) \text{ in the worst case}] = \Pr[Y_q < \lambda \ln(n + 1) \text{ for all } q \in Q] = 1 - \Pr[Y_q \geq \lambda \ln(n + 1) \text{ for some } q \in Q] \leq 1 - \sum_{q \in Q} \Pr[Y_q \geq \lambda \ln(n + 1)] \geq 1 - |Q| \left(\frac{1}{n + 1}\right)^{\ln(5/4) - 1} \geq 1 - 2(n + 1)^2 \cdot \left(\frac{1}{n + 1}\right)^3 = 1 - \frac{2}{n + 1}.$$