(a) Es gilt: $p_A \geq \frac{1}{\binom{n}{2}}$
$\Rightarrow$ Der Erwartungswert der #Wiederholungen, bis wir das erste Mal den Minimalen Schnitt ausgeben ist höchstens $\binom{n}{2}$
$\Rightarrow$ Unser Algorithmus $A\_IMPROVED$ ruft A $\lambda\binom{n}{2}$ mal für ein $\lambda > 0$ und geben dann der kleinste je erhaltenen Wert aus.
Die Wahrscheinlichkeit dass wir nach $\lambda\binom{n}{2}$ Mal nicht einen minimalen Schnitt finden ist höchstens:

$$(1-p_A)^{\lambda\binom{n}{2}} \leq (1-\frac{1}{\binom{n}{2}})^{\lambda\binom{n}{2}} \leq e^{-\lambda}$$

$\Rightarrow$ Der kleinste angetroffene Wert ist mit einer Wahrscheinlichkeit von mindestens $1-e^{-\lambda}$
Wir wollen dass der kleinste angetroffene Wert mit einer Wahrscheinlichkeit von mindestens 99% der Minimalen schnitt ist.
$\Rightarrow 0.99 = 1-e^{-\lambda} \to 0.01 = e^{-\lambda} \to e^{\lambda} = 100 \to \lambda = ln(100)$
$\Rightarrow$ $A\_IMPROVED$ ruft A $\lceil ln(100)\binom{n}{2}\rceil$ mal und gibt der kleinste je erhaltenen Schnitt aus.

(b) Der algorithmus B ist ein Monte-Carlo Algorithmus mit Einseitiger Fehler mit:

$$\epsilon \geq \frac{1}{2}$$
$$\delta \leq 0.01$$

Unser Algorithmus $B\_IMPROVED$ ruft B solange auf bis entweder Nein ausgegeben wird oder $N = \epsilon^{-1}ln(\delta^{-1})$ mal Ja ausgegeben wurde.
$\Rightarrow$ $N \leq 2 \cdot ln(100)$ Nach satz 2.74 gibt unser algorithmus dass richtige ergebnis mit einer wahrscheinlichkeit Pr[korrekt]$\geq$ 0.99

(c) Der Algorithmus C ist ein Monte-Carlo Algorithmus mit Beidseitiger Fehler wobei:

$$\epsilon \geq \frac{3}{4} - \frac{1}{2} = \frac{1}{4}$$
$$\delta \leq 0.01$$

Unser Algorithmus $C\_IMPROVED$ macht $N = 4\epsilon^{-2}ln(\delta^{-1})$ unabhängige Aufrufe und gibt die Mehrheit der erhaltenen Antworten aus.
$\Rightarrow$ $N = \lceil 64ln(100)\rceil$ Nach satz 2.75 gibt unser algorithmus dass richtige ergebnis mit einer wahrscheinlichkeit Pr[korrekt]$\geq$ 0.99

(d) Ziel: Gegeben ein Graph G mit B=$|V|-1$, wir wollen entscheiden, ob es in G einen Pfad der Länge B gibt. Dazu färben wir die Knoten zufällig mit den Farben [k] (k:= B+1) und prüfen, ob es einen bunten Pfad der Länge k-1 gibt. Angenommen G enthält einen Pfad der Länge k-1,aus dem Skript (satz 3.2) wissen wir die Erfolgswahrscheinlichkeit ein solcher Pfad P zu finden ist:

$$Pr_{Erfolg} := Pr[\exists \text{ bunter Pfad der Länge k-1}] \geq Pr[\text{ P ist bunt }] = \frac{k!}{k^k} \geq e^{-k}$$

Da wir ein geometrische Verteilung haben ist der Erwartungswert der Anzahl Versuche bis man einen bunten Pfad der Länge k-1 bei wiederholten Färbungen mit k Farben $\frac{1}{Pr_{Erfolg}} \leq e^k$ (satz 3.2 Teil 2)
Wir bauen jetzt einen Monte Carlo Algorithmus. Dazu wählen wir ein $\lambda \in \mathbb{R}$ mit $\lambda > 1$ und wiederholen unseren Test höchstens $\lceil \lambda e^k \rceil$ mal, bis wir eine Bestätigung haben, dass es einen Pfad der Länge k-1 gibt. Gelingt dies, antworten wir Ja, wenn es in allen Versuchen scheitert antworten wir Nein.
$\Rightarrow$ Antwortet der Algorithmus mit Ja, dann hat der Graph einen Pfad der Länge k-1. Hat der Graph einen Pfad der Länge k-1, dann ist die Wahrscheinlichkeit, dass der Algorithmus mit Nein antwortet höchstens $e^{-\lambda}$ (folgt aus der gleiche begründung wie in a)). Laut Aufgabestellung wollen wir, dass dies mit Wahrscheinlichkeit $\frac{1}{2}$ passiert.
$\Rightarrow e^{\lambda} = 2 \to \lambda = ln(2) \Rightarrow$ Unser Monte Carlo Algorithmus hat nach Satz 3.3 eine Laufzeit von $\mathcal{O}(\lambda(2e)^k)km) = \mathcal{O}((2e)^k km)$ und liefert mit Wahrscheinlichkeit von $1-e^{-\lambda} = \frac{1}{2}$ die korrekte Antwort.
Um den Wahrscheinlichkeit auf eine korrekte Antwort auf mindestens 99% zu bringen nutzen wir den Prinizip aus teilaufgabe b). Wir nehmen unsere Algorithmus von oben und bezeichnen es mit D. D ist ein Monte-Carlo Algorithmus mit Einseitiger Fehler:

$$\epsilon \geq \frac{1}{2}$$
$$\delta \leq 0.01$$

Wir machen einen Algorithmus $D\_IMPROVED$ welche der Algorithmus D als subroutine aufruft bis entweder Nein ausgegeben wird oder $N = \epsilon^{-1}ln(\delta^{-1}) = \lceil 2 \cdot ln(100)\rceil$ mal Ja ausgegeben wurde. Nach satz 2.74 gibt usner algorithmus dass richtige ergebnis mit einer Wahrscheinlichkeit $Pr[korrekt] \geq 0.99$