1. The algorithm chooses a random vector $\vec{r} \in \{0,1\}^n$, where each entry in $\vec{r}$ is chosen uniformly at random to be 0 or 1. We output YES if $AB\vec{r} = C\vec{r}$ and NO otherwise. The time complexity is $O(n^2)$, since we can first calculate $B\vec{r}$ and then $A(B\vec{r})$.

Clearly if $AB = C$ then $AB\vec{r} = C\vec{r}$ no matter what $\vec{r}$ is. We claim that if $AB \neq C$ then $\Pr[AB\vec{r} = C\vec{r}] \leq 1/2$. The analysis we present here is from the book [1].

Let $D = AB - C$. We know that $D$ is not the all zeroes matrix. We wish to bound the probability that $D\vec{r} = 0$. Without loss of generality, we may assume that the first row in $D$ has a non-zero entry, and that all the non-zero entries in that row precede the zero entries. Let $\vec{d}$ be the vector consisting of the entries from the first row in $D$, and assume that the first $k > 0$ entries in $\vec{d}$ are non-zero. Now an upper bound on the probability that $D\vec{r} = 0$ is the probability that the first entry, $\vec{d} \cdot \vec{r} = 0$. Now $\vec{d} \cdot \vec{r} = 0$ if and only if

$$r_1 = -\sum_{i=2}^{k} \frac{d_i r_i}{d_1}$$

Now assuming that the right hand side is a fixed value, the random value $r_1$ can be equal to it with probability at most 1/2.

2. Rabin’s closest pairs algorithm will essentially remain the same in three dimensions. At each step we will maintain that no cell in the (now 3-dimensional) grid with side $r$ has more than 9 points, because otherwise the distance between two of the points will be smaller than $r$. We now will have 26 adjacent cells instead of 8. Other details [2] remain the same and we still end up with an expected linear time algorithm.

3. We show below an example where the analysis is tight. Any matching derived vertex cover will have size 2, whereas the minimum vertex cover has size 1 (the center vertex).

![Figure 1: A star graph.](image)

4. We will use the same algorithm that we used for set cover: greedily pick a set that covers the maximum number of uncovered elements and repeat $k$ times.

The analysis will also be similar. Let $OPT$ denote the maximum number of covered elements in the optimal solution. After step $i$, let $x_i$ denote the number of new elements covered, $y_i$ the number of elements covered so far and let $z_i = OPT - y_i$. Then $x_{i+1} \geq z_i/k$ since at each step, some set must
cover at least $1/k$ fraction of the remaining uncovered elements from $OPT$. Now since $z_{i+1} \leq z_i - x_{i+1}$ we get

$$z_{i+1} \leq \left(1 - \frac{1}{k}\right) z_i \leq \left(1 - \frac{1}{k}\right)^{i+1} OPT.$$  

At the end, $z_k \leq \left(1 - \frac{1}{e}\right)^k OPT \leq OPT/e$. Hence the total number of covered elements,

$$y_k = OPT - z_k \geq (1 - 1/e)OPT.$$  

References
