The Forehead Game

Exposition by William Gasarch

July 25, 2022

The Problem

Alice is A, Bob is B, Carol is C.

The Problem

Alice is A, Bob is B, Carol is C.

1. A, B, and C have a string of length n on their foreheads.

The Problem

Alice is A, Bob is B, Carol is C.

1. A, B, and C have a string of length n on their foreheads.
2. Foreheads: A's has a; B's has b; C's has c.

The Problem

Alice is A, Bob is B, Carol is C.

1. A, B, and C have a string of length n on their foreheads.
2. Foreheads: A's has a; B's has b; C's has c.
3. A knows b, c; B knows a, c; C knows a, b.

The Problem

Alice is A, Bob is B, Carol is C.

1. A, B, and C have a string of length n on their foreheads.
2. Foreheads: A's has a; B's has b; C's has c.
3. A knows b, c; B knows a, c; C knows a, b.
4. They want to know if $a + b + c = 2^{n+1} - 1$.

The Problem

Alice is A, Bob is B, Carol is C.

1. A, B, and C have a string of length n on their foreheads.
2. Foreheads: A's has a; B's has b; C's has c.
3. A knows b, c; B knows a, c; C knows a, b.
4. They want to know if $a + b + c = 2^{n+1} - 1$.
5. **Solution** A says b, B then computes $a + b + c$ and then says YES if $a + b + c = 2^{n+1} - 1$, NO if not.

The Problem

Alice is A, Bob is B, Carol is C.

1. A, B, and C have a string of length n on their foreheads.
2. Foreheads: A's has a;   B's has b;   C's has c.
3. A knows b, c;   B knows a, c;   C knows a, b.
4. They want to know if $a + b + c = 2^{n+1} - 1$.
5. **Solution** A says b, B then computes $a + b + c$ and then says YES if $a + b + c = 2^{n+1} - 1$, NO if not.
6. **Solution** uses $n + 1$ bits of comm. Can do better?

Vote

Vote

1. Any protocol requires $n+1$ bits, hence the one given that takes $n+1$ is the best you can do.

Vote

1. Any protocol requires $n+1$ bits, hence the one given that takes $n+1$ is the best you can do.
2. There is a protocol that takes αn bits for some $\alpha < 1$ but any protocol requires $\Omega(n)$ bits.

Vote

1. Any protocol requires $n+1$ bits, hence the one given that takes $n+1$ is the best you can do.
2. There is a protocol that takes αn bits for some $\alpha < 1$ but any protocol requires $\Omega(n)$ bits.
3. There is a protocol that takes $\ll n$ bits.

Vote

1. Any protocol requires $n+1$ bits, hence the one given that takes $n+1$ is the best you can do.
2. There is a protocol that takes αn bits for some $\alpha < 1$ but any protocol requires $\Omega(n)$ bits.
3. There is a protocol that takes $\ll n$ bits.

STUDENTS WORK IN GROUPS TO BEAT $n+1$ OR SHOW YOU CAN"T

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \ \cdots \ , \ c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \cdots, c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.
3. Bob knows c_i's so he now knows $b_{n/2}, \ldots, b_{n-1}$.

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \cdots, c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.
3. Bob knows c_i's so he now knows $b_{n/2}, \ldots, b_{n-1}$.
   Bob knows a_i's and c_i's so he can compute
   $a_{n/2} \cdots a_{n-1} + b_{n/2} \cdots b_{n-1} + c_{n/2} \cdots c_{n-1} = s + \text{carry } z$

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$,   B:$b_0 \cdots b_{n-1}$,   C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \;\cdots\;, \; c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.
3. Bob knows c_i's so he now knows $b_{n/2}, \ldots, b_{n-1}$.
   Bob knows a_i's and c_i's so he can compute
   $a_{n/2} \cdots a_{n-1} + b_{n/2} \cdots b_{n-1} + c_{n/2} \cdots c_{n-1} = s + \text{carry } z$
   $s = 1^{n/2}$: Bob says (MAYBE,z). $s \neq 1^{n/2}$: Bob says NO.

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \; \cdots \;, \; c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.
3. Bob knows c_i's so he now knows $b_{n/2}, \ldots, b_{n-1}$.
   Bob knows a_i's and c_i's so he can compute
   $a_{n/2} \cdots a_{n-1} + b_{n/2} \cdots b_{n-1} + c_{n/2} \cdots c_{n-1} = s + \text{carry } z$
   $s = 1^{n/2}$: Bob says (MAYBE,z). $s \neq 1^{n/2}$: Bob says NO.
4. Carol knows b_i's so she now knows $c_0, \ldots, c_{n/2-1}$.

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \ \cdots \ , \ c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.
3. Bob knows c_i's so he now knows $b_{n/2}, \ldots, b_{n-1}$.
   Bob knows a_i's and c_i's so he can compute
   $a_{n/2} \cdots a_{n-1} + b_{n/2} \cdots b_{n-1} + c_{n/2} \cdots c_{n-1} = s + \text{carry } z$
   $s = 1^{n/2}$: Bob says (MAYBE,z). $s \neq 1^{n/2}$: Bob says NO.
4. Carol knows b_i's so she now knows $c_0, \ldots, c_{n/2-1}$.
   Carol knows the carry bit z so she can compute
   $a_0 \cdots a_{n/2} + b_0 \cdots b_{n/2} + c_0 \cdots c_{n/2} + z = t$

Protocol in $\frac{n}{2} + O(1)$ bits Due to Dean Foster

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$.
2. A says: $c_0 \oplus b_{n/2}, \; \cdots \;, \; c_{n/2-1} \oplus b_{n-1}$. $n/2$ bits.
3. Bob knows c_i's so he now knows $b_{n/2}, \ldots, b_{n-1}$.
   Bob knows a_i's and c_i's so he can compute
   $a_{n/2} \cdots a_{n-1} + b_{n/2} \cdots b_{n-1} + c_{n/2} \cdots c_{n-1} = s + \text{carry } z$
   $s = 1^{n/2}$: Bob says (MAYBE,z). $s \neq 1^{n/2}$: Bob says NO.
4. Carol knows b_i's so she now knows $c_0, \ldots, c_{n/2-1}$.
   Carol knows the carry bit z so she can compute
   $a_0 \cdots a_{n/2} + b_0 \cdots b_{n/2} + c_0 \cdots c_{n/2} + z = t$
   $t = 1^{n/2}$: Carol says YES. $t \neq 1^{n/2}$: Carol says NO.

Four People

Alice is A, Bob is B, Carol is C, Donna is D.

Four People

Alice is A, Bob is B, Carol is C, Donna is D.

1. A, B, C, D have a string of length n on their foreheads.

Four People

Alice is A, Bob is B, Carol is C, Donna is D.

1. A, B, C, D have a string of length n on their foreheads.
2. A's forehead has a, B's forehead has b, ...

Four People

Alice is A, Bob is B, Carol is C, Donna is D.

1. A, B, C, D have a string of length n on their foreheads.
2. A's forehead has a, B's forehead has b, ...
3. They want to know if $a + b + c + d = 2^{n+1} - 1$.

Four People

Alice is A, Bob is B, Carol is C, Donna is D.

1. A, B, C, D have a string of length n on their foreheads.
2. A's forehead has a, B's forehead has b, ...
3. They want to know if $a+b+c+d=2^{n+1}-1$.
4. **Obvious Solution** uses $n+1$ bits of comm. Can do better?

Four People

Alice is A, Bob is B, Carol is C, Donna is D.

1. A, B, C, D have a string of length n on their foreheads.
2. A's forehead has a, B's forehead has b, ...
3. They want to know if $a + b + c + d = 2^{n+1} - 1$.
4. **Obvious Solution** uses $n + 1$ bits of comm. Can do better?

STUDENTS WORK IN GROUPS TO EITHER DO BETTER THAN $n + 1$ OR SHOW YOU CAN"T

Protocol for 4 in $\frac{n}{3} + O(1)$ bits

Protocol for 4 in $\frac{n}{3} + O(1)$ bits

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$, D:$d_0 \cdots d_{n-1}$.

Protocol for 4 in $\frac{n}{3} + O(1)$ bits

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$, D:$d_0 \cdots d_{n-1}$.
2. A says: $c_0 \oplus b_{n/3-1} \oplus d_{2n/3-1}, \cdots, c_{n/3-1} \oplus b_{2n/3-1} \oplus d_{n-1}$.

Protocol for 4 in $\frac{n}{3} + O(1)$ bits

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$, D:$d_0 \cdots d_{n-1}$.
2. A says: $c_0 \oplus b_{n/3-1} \oplus d_{2n/3-1}, \cdots, c_{n/3-1} \oplus b_{2n/3-1} \oplus d_{n-1}$.
3. Carol can add first 1/3 of the bits, sees if its $1^{n/3}$, if its not say NO, if it is say MAYBE and the carry bit.

Protocol for 4 in $\frac{n}{3} + O(1)$ bits

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$, D:$d_0 \cdots d_{n-1}$.
2. A says: $c_0 \oplus b_{n/3-1} \oplus d_{2n/3-1}, \ \cdots \ , \ c_{n/3-1} \oplus b_{2n/3-1} \oplus d_{n-1}$.
3. Carol can add first $1/3$ of the bits, sees if its $1^{n/3}$, if its not say NO, if it is say MAYBE and the carry bit.
4. Bob can add second $1/3$ of the bits along with the carry bit, sees if its $1^{n/3}$, if its not say NO, if it is say MAYBE and the carry bit.

Protocol for 4 in $\frac{n}{3} + O(1)$ bits

1. A:$a_0 \cdots a_{n-1}$, B:$b_0 \cdots b_{n-1}$, C:$c_0 \cdots c_{n-1}$, D:$d_0 \cdots d_{n-1}$.
2. A says: $c_0 \oplus b_{n/3-1} \oplus d_{2n/3-1}, \; \cdots \;, \; c_{n/3-1} \oplus b_{2n/3-1} \oplus d_{n-1}$.
3. Carol can add first $1/3$ of the bits, sees if its $1^{n/3}$, if its not say NO, if it is say MAYBE and the carry bit.
4. Bob can add second $1/3$ of the bits along with the carry bit, sees if its $1^{n/3}$, if its not say NO, if it is say MAYBE and the carry bit.
5. Donna can add third $1/3$ of the bits along with the carry bit, sees if its $1^{n/3}$, if its not say NO, if it is say YES.

k People

People are $A_1, \ldots, A_k$.

k People

People are $A_1, \ldots, A_k$.

1. A_i has a string of length n on their foreheads.

k People

People are $A_1, \ldots, A_k$.

1. A_i has a string of length n on their foreheads.
2. A_i's forehead has a_i

k People

People are $A_1, \ldots, A_k$.

1. A_i has a string of length n on their foreheads.
2. A_i's forehead has a_i
3. They want to know if $a_1 + \cdots + a_k = 2^{n+1} - 1$.

k People

People are $A_1, \ldots, A_k$.

1. A_i has a string of length n on their foreheads.
2. A_i's forehead has a_i
3. They want to know if $a_1 + \cdots + a_k = 2^{n+1} - 1$.
4. Can do in $\frac{n}{k-1} + O(1)$ bits, similar to the $k = 3, 4$ cases.

k People

People are $A_1, \ldots, A_k$.

1. A_i has a string of length n on their foreheads.
2. A_i's forehead has a_i
3. They want to know if $a_1 + \cdots + a_k = 2^{n+1} - 1$.
4. Can do in $\frac{n}{k-1} + O(1)$ bits, similar to the $k = 3, 4$ cases.
5. Caveat: The $O(1)$ term is really $O(k)$ which matters if k is a function of n.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

1. $\frac{n}{2} + O(1)$ is roughly optimal.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

1. $\frac{n}{2} + O(1)$ is roughly optimal.
2. There is an $O\left(\frac{n}{\log n}\right)$ protocol and it is roughly optimal.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

1. $\frac{n}{2} + O(1)$ is roughly optimal.
2. There is an $O\left(\frac{n}{\log n}\right)$ protocol and it is roughly optimal.
3. There is an $O\left(\frac{n}{\log n}\right)$ protocol, optimal UNKNOWN.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

1. $\frac{n}{2} + O(1)$ is roughly optimal.
2. There is an $O\left(\frac{n}{\log n}\right)$ protocol and it is roughly optimal.
3. There is an $O\left(\frac{n}{\log n}\right)$ protocol, optimal UNKNOWN.
4. There exists an $O(n^{1/2})$ protocol and it is roughly optimal.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

1. $\frac{n}{2} + O(1)$ is roughly optimal.
2. There is an $O\left(\frac{n}{\log n}\right)$ protocol and it is roughly optimal.
3. There is an $O\left(\frac{n}{\log n}\right)$ protocol, optimal UNKNOWN.
4. There exists an $O(n^{1/2})$ protocol and it is roughly optimal.
5. There exists an $O(n^{1/2})$ protocol, optimal UNKNOWN.

Final Vote and the Answer

Lets go back to 3 people. We know we can do $\frac{n}{2} + O(1)$.

1. $\frac{n}{2} + O(1)$ is roughly optimal.
2. There is an $O\left(\frac{n}{\log n}\right)$ protocol and it is roughly optimal.
3. There is an $O\left(\frac{n}{\log n}\right)$ protocol, optimal UNKNOWN.
4. There exists an $O(n^{1/2})$ protocol and it is roughly optimal.
5. There exists an $O(n^{1/2})$ protocol, optimal UNKNOWN.

VOTE!

k people Answer and Some History

3 people:

k people Answer and Some History

3 people:

- Chandra-Furst-Lipton (CFL) (1983): $O(n^{1/2})$ protocol.
  `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/mpp.pdf`,
  `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/expositionofCFG.pdf`

k people Answer and Some History

3 people:

- Chandra-Furst-Lipton (CFL) (1983): $O(n^{1/2})$ protocol. `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/mpp.pdf`, `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/expositionofCFG.pdf`
- They needed this lemma to get lower bounds in computer science. Better lower bounds were later proven using easier techniques. However, by then **The Forehead Problem** had taken on a life of its own.

k people Answer and Some History

3 people:

- Chandra-Furst-Lipton (CFL) (1983): $O(n^{1/2})$ protocol. `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/mpp.pdf`, `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/expositionofCFG.pdf`
- They needed this lemma to get lower bounds in computer science. Better lower bounds were later proven using easier techniques. However, by then **The Forehead Problem** had taken on a life of its own.
- CFL: constructive and did not have the constants. Linial and Shraibman: explicitly protocol that uses $n^{1/2} + o(n^{1/2})$ bits. See `https://arxiv.org/pdf/2102.00421.pdf`

k people Answer and Some History

3 people:

- Chandra-Furst-Lipton (CFL) (1983): $O(n^{1/2})$ protocol. `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/mpp.pdf`, `https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/expositionofCFG.pdf`
- They needed this lemma to get lower bounds in computer science. Better lower bounds were later proven using easier techniques. However, by then **The Forehead Problem** had taken on a life of its own.
- CFL: constructive and did not have the constants. Linial and Shraibman: explicitly protocol that uses $n^{1/2} + o(n^{1/2})$ bits. See `https://arxiv.org/pdf/2102.00421.pdf`
- CFL showed lower bound $\Omega(1)$.

k people Answer and Some History

3 people:

- Chandra-Furst-Lipton (CFL) (1983): $O(n^{1/2})$ protocol. https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/mpp.pdf, https://www.cs.umd.edu/~gasarch/TOPICS/ramsey/expositionofCFG.pdf
- They needed this lemma to get lower bounds in computer science. Better lower bounds were later proven using easier techniques. However, by then **The Forehead Problem** had taken on a life of its own.
- CFL: constructive and did not have the constants. Linial and Shraibman: explicitly protocol that uses $n^{1/2} + o(n^{1/2})$ bits. See https://arxiv.org/pdf/2102.00421.pdf
- CFL showed lower bound $\Omega(1)$.
- Gasarch (2006): Lower Bound $\Omega(\log \log n)$.

k people Answer and Some History

k people Answer and Some History

- Gasarch 2006: there is an $O(n^{1/(\log_2(k-1))})$ protocol. (A more careful analysis of CFL protocol.)

k people Answer and Some History

- Gasarch 2006: there is an $O(n^{1/(\log_2(k-1))})$ protocol. (A more careful analysis of CFL protocol.)
- CFL lower bound $\Omega(1)$.

k people Answer and Some History

- Gasarch 2006: there is an $O(n^{1/(\log_2(k-1))})$ protocol. (A more careful analysis of CFL protocol.)
- CFL lower bound $\Omega(1)$.
- Nothing else is known.

Open Problem

For 3 people we have:

Open Problem

For 3 people we have:

1. Elementary proof: Protocol $\frac{n}{2} + O(1)$.

Open Problem

For 3 people we have:

1. Elementary proof: Protocol $\frac{n}{2} + O(1)$.
2. Hard proof: Protocol $O(n^{1/2})$.

Open Problem

For 3 people we have:

1. Elementary proof: Protocol $\frac{n}{2} + O(1)$.
2. Hard proof: Protocol $O(n^{1/2})$.

Open Find an elementary proof for a protocol, $< \frac{n}{2} + O(1)$.

Open Problem

For 3 people we have:

1. Elementary proof: Protocol $\frac{n}{2} + O(1)$.
2. Hard proof: Protocol $O(n^{1/2})$.

Open Find an elementary proof for a protocol, $< \frac{n}{2} + O(1)$.

Open Similar questions for 4 people, 5 people, etc.